Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung

On the Stability of Distribution Topologies in Peer-to-Peer Live Streaming Systems

Dissertation

zur Erlangung des Akademischen Grades
Dr. rerum naturalium (Dr. rer. nat.)

---

vorgelegt von: Sascha Grau
geboren am: 29.03.1982 in Erfurt
Datum der Einreichung: 04.06.2012
Datum der Verteidigung: 08.11.2012

1. Gutachter: Prof. Dr. rer. nat. habil. Manfred Kunde
2. Gutachter: Prof. Dr.-Ing. Günter Schäfer
3. Gutachter: Prof. Dr. rer. nat. habil. Christian Schindelhauer

urn:nbn:de:gbv:ilm1-2012000395

Abstract

In the recent years, live streaming of multimedia content has developed into a major application on the Internet. Following a peer-to-peer approach, it can be realized despite a limited availability of computing and bandwidth resources at the data source. For this, the source and the receivers (called peers) cooperate to distribute the data to all participants.

Such peer-to-peer live streaming systems support the freedom of information. Furthermore, they provide a powerful and cost-efficient alternative to the traditional distribution channels of live media content.

However, the stability of peer-to-peer live streaming systems is constantly challenged. Especially, the unreliability and vulnerability of their participants allows for failures and attacks suddenly disabling certain sets of peers. The consequences of such events on a streaming system are largely determined by its distribution topology. The latter reflects the pattern of communication between the system's participants.

In this thesis, we analyze a broad range of optimization problems occuring when modeling different notions of stability of such distribution topologies. Besides obtaining insights that are directly applicable in the design of stable peer-to-peer live streaming systems, we also identify connections with very different areas of mathematics.

When measuring damage dealt to a streaming system, we account for system-wide packet loss as well as the decrease in streaming quality perceived by individual peers. We discuss notions of stability against both attacks and failures. However, we set a special focus on attack-stability. Here, we follow two different approaches.

At first, we investigate the computational complexity and approximability of the problem of finding resource-efficient attacks creating a given amount of damage. This allows to point out computational limitations for an attacker's planning abilities. Additionally, it demonstrates the influence of the chosen stream encoding and important topology properties on the hardness of such attack problems.

Then, we turn to study topology formation problems. Here, a set of topology parameters is given and the task consists in finding an eligible distribution topology. In particular, this topology has to minimize the maximum damage achievable by attacks with arbitrary attack parameters.

We identify necessary and sufficient conditions on attack-stable distribution topologies. Thereby, we give mathematically sound guidelines for the topology management of peer-to-peer live streaming systems. We find large classes of efficiently-constructable topologies minimizing the system-wide packet loss under attacks. Additionally, we show that determining this feature for arbitrary topologies is **coNP**-complete.

Considering topologies minimizing the maximum number of peers for which an attack leads to a heavy decrease in perceived streaming quality, the requirements change.

Here, we show that the corresponding topology formation problem is closely related to long-standing open problems of Design and Coding Theory.

Finally, we study the formation problem for topologies that are stable against uncoordinated failures. We develop a probabilistic failure model and study topologies minimizing the expected system-wide packet loss. We identify necessary and sufficient conditions, and show that the existence of such topologies depends on the bandwidth available to the source and the peers. In the case of their existence, we prove that it is **NP**-complete to find failure-stable topologies.

The identified conditions on stable distribution topologies give possible optimization goals for the topology management of peer-to-peer live streaming systems. The demonstrated limitations, both for attackers and the efficient construction of certain classes of stable topologies, can support the evaluation of trade-offs between threat and costs of safeguarding. Consequently, they help in choosing appropriate stability goals that match a streaming system's intended use.

Zusammenfassung

Die Verteilung live gesendeter Multimedia-Inhalte über das Internet hat in den letzten Jahren zunehmend an Bedeutung gewonnen. Wird sie mittels eines Peer-to-Peer-Ansatzes realisiert, sind an der Quelle des Datenstroms keine umfangreichen Berechnungs- und Bandbreitenresourcen notwendig. Stattdessen kooperieren die Quelle und die Empfänger (Peers genannt), um die Daten an alle Teilnehmer zu verteilen.

Derartige Peer-to-Peer Live-Streaming-Systeme bieten neue Möglichkeiten zur Sicherstellung der Informationsfreiheit. Außerdem stellen sie eine leistungsfähige und kostengünstige Alternative zu traditionellen Verteilungswegen für Medieninhalte dar.

Peer-to-Peer Live-Streaming-Systeme sind jedoch ständigen Störungen ausgesetzt. Insbesondere können unzuverlässige und leicht angreifbare Teilnehmer Ausfälle und Angriffe ermöglichen, welche überraschend bestimmte Teilmengen von Peers aus dem System entfernen. Die Folgen solcher Vorfälle werden großteils von der verwendeten Verteilungstopologie bestimmt. Diese bildet die Kommunikationsverbindungen zwischen den Teilnehmern des Streaming-Systems ab.

In dieser Arbeit analysieren wir eine breite Palette von Optimierungsproblemen welche bei der Betrachtung verschiedener Stabilitätsbegriffe für solche Verteilungstopologien auftreten. Hierdurch gelangen wir zu zahlreichen Erkenntnissen die direkt in das Design stabiler Peer-to-Peer Live-Streaming-Systeme einfließen können. Zusätzlich identifizieren wir Verbindungen zu sehr verschiedenen Gebieten der Mathematik.

Bei der Messung des an einem Streaming System auftretenden Schadens berücksichtigen wir sowohl systemweite Paketverluste als auch das Absinken der von einzelnen Teilnehmern wahrgenommenen Stream-Qualität. Wir untersuchen Stabilitätsbegriffe im Falle von Angriffen und bei Auftreten von Ausfällen. Einen besonderen Schwerpunkt setzen wir jedoch auf die Stabilität gegen Angriffe. Hierbei werden wir zwei verschiedene Ansätze verfolgen.

Zunächst untersuchen wir die Berechnungskomplexität und Approximierbarkeit des Problems einen resourcen-effizienten Angriff zu finden, welcher einen vorgegebenen Schadenswert erreicht. Ein solcher Ansatz erlaubt es grundsätzliche Beschränkungen in den Planungsmöglichkeiten von Angreifern zu identifizieren. Zusätzlich können wir zeigen, wie die gewählte Datenstromkodierung und wichtige Topologieparameter die Schwierigkeit solcher Angriffsprobleme beeinflussen.

Anschließend studieren wir Topologieformationsprobleme. Dabei sind bestimmte Topologieparameter vorgegeben und es muss eine passende Verteilungstopologie gefunden werden. Ziel ist es Topologien zu erzeugen, welche den durch Angriffe mit beliebigen Parametern erzeugbaren maximalen Schaden minimieren.

Wir identifizieren notwendige und hinreichende Eigenschaften solcher angriffsstabilen Verteilungstopologien. Hierdurch lassen sich mathematisch fundierte Zielstellungen für das Topologie-Management von Peer-to-Peer Live-Streaming-Systemen geben. Wir

zeigen zwei große Klassen effizient konstruierbarer Verteilungstopologien, welche den maximal möglichen, durch Angriffe verursachten Paketverlust minimieren. Zusätzlich beweisen wir, dass die Bestimmung dieser Eigenschaft für beliebige Topologien **coNP**-vollständig ist. Soll die maximale Anzahl von Peers minimiert werden, bei denen ein Angriff zu stark verminderter Stream-Qualität führt, ändern sich die Anforderungen an Verteilungstopologien. Wir zeigen, dass das korrespondierende Topologieformationsproblem eng mit offenen Problemen aus Design- und Kodierungstheorie verwandt ist.

Zusätzlich befassen wir uns mit dem Formationsproblem für Verteilungstopologien die stabil in Bezug auf unkoordiniert auftretende Ausfälle sind. Hier entwickeln wir ein probabilistisches Ausfallmodell und setzen das Ziel den erwarteten Paketverlust zu minimieren. Wir identifizieren notwendige und hinreichende Bedingungen an entsprechende Verteilungstopologien und zeigen dass ihre Existenz von der Bandbreite abhängt, welche der Stromquelle und den Peers zur Verfügung steht. Existieren ausfallstabile Topologien, so beweisen wir die **NP**-Vollständigkeit des entsprechenden Topologieformationsproblems.

Die in dieser Arbeit identifizierten Bedingungen an stabile Verteilungstopologien geben mögliche Zielstellungen für das Topologie-Management von Peer-to-Peer Live-Streaming-Systemen vor. Die gezeigten Beschränkungen für Angreifer und effiziente Topologiekonstruktion ermöglichen es Bedrohungen und Absicherungsaufwand abzuwiegen. Sie helfen somit für den jeweiligen Anwendungsfall geeignete Stabilitätsziele auszuwählen.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich während der Entstehung dieser Arbeit begleitet haben.

Eine wichtige Rolle hat dabei mein Betreuer Prof. Dr. Manfred Kunde eingenommen. Er hat mir während meiner gesamten Promotionsphase viel Vertrauen geschenkt und große wissenschaftliche Freiheiten gewährt. Ich hoffe die Ergebnisse bestätigen ihn in diesem Vorgehen.

Sehr profitiert habe ich auch von Prof. Dr. Günter Schäfer mit seinen vielzähligen Anregungen und Ideen. Außerdem sorgte er für die gelegentlich notwendigen Impulse wissenschaftliche Ergebnisse in Publikationen zu verwandeln.

Neben diesen beiden Gutachtern danke ich ebenso Prof. Dr. Christian Schindelhauer für die Bereitschaft sich intensiv mit meiner Arbeit auseinanderzusetzen. Seine detaillierten und stets konstruktiven Anregungen waren ausgesprochen hilfreich.

Desweiteren möchte ich mich auch bei Prof. Dr. Martin Dietzfelbinger und Prof. Dr. Dietrich Kuske bedanken. Beide waren als Ansprechpartner immer für mich da. Ich habe die gute Zusammenarbeit sowie die offene und freundliche Atmosphäre in unserem Institut stets als sehr angenehm empfunden.

Ein wesentlicher Teil dieser Atmosphäre wurde auch von Petra Schüller und Jana Kopp geprägt. Sie organisierten und kümmerten sich um all die kleinen und großen Probleme des Alltags. Ich danke ihnen nicht nur dafür.

Ich danke Prof. Dr. Thorsten Strufe und Dr. Michael Brinkmeier, da sie die Grundlagen für meine Dissertation gelegt haben. Insbesondere Michael hat mich erst in die Position gebracht, mich diesem interessanten Thema widmen zu können. Er hat an mich geglaubt und war immer ein guter Freund. Danke.

Außerdem bin ich sehr glücklich mit meinen großartigen Kollegen zusammenarbeiten zu dürfen. Michael Rink, Christopher Mattern, Dr. Mathias Fischer, Dr. Michael Rossberg, Dr. Ulf Schellbach, Martin Huschenbett, Roy Mennicke und Martin Aumüller schlugen sich an meiner Seite durch die Höhen und Tiefen des Wissenschaftsbetriebs. Insbesondere Martin Aumüller ist mir darüber hinaus zu einem guten Freund geworden, der sich nie scheut auch unbequeme Wahrheiten auszusprechen.

Besonders wichtig war eine solche Eigenschaft während des Korrekturlesens dieser Arbeit. Für ihren Einsatz bei dieser wenig reizvollen Aufgabe danke ich außerdem Dr. Michael Rossberg, Dr. Mathias Fischer, Martin Niebergall, Stephan Beyer und Claudia Oberländer.

Nicht vergessen möchte ich zusätzlich Andreas Brieg, Vincent Holluba, Johannes Röckert und Wolfgang Gummlich, die in ihren Abschlussarbeiten sowie ihren HiWi-Tätigkeiten einen Teil des Weges mit mir gegangen sind, und die den Enthusiasmus an meiner Forschung geteilt haben.

Zuletzt bleibt festzustellen, dass ich diese Arbeit nur mit Hilfe meiner Familie realisieren konnte. Bei allen Rückschlägen und Fragezeichen, die über die Jahre aufgetreten sind, war sie stets das unverrückbare Fundament an dem ich mich festhalten konnte. Aus diesem Grund danke ich meinen Eltern für dreißig Jahre bedingungslose Unterstützung und Vertrauen.

Vor allem aber möchte ich mich bei meiner Frau Katrin und meinen wundervollen Kindern bedanken. Sie haben mich jeden Tag aufs Neue daran erinnert, dass es im Leben wesentlich wichtigere Dinge gibt als wissenschaftliche Fragestellungen. Sie haben mir all den Halt und die Kraft gegeben, die notwendig waren um diese Arbeit entstehen zu lassen.

Contents

1. Introduction

1.1. Motivation and Goals

A peer-to-peer-based approach to distribute live multimedia data over the Internet offers a scalable, cost-efficient, and powerful alternative to classical systems of media distribution. It was adopted by [CRSZ02, CDK+03, MR07, BSS09] and many more. Instead of relying only on pairwise communication between a central source and each client, such systems exploit the resources of their participants (named *peers*), which actively take part in the distribution process. Consequently, peer-to-peer systems promise to supply the multimedia content to a number of participants that would otherwise far exceed the bandwidth resources of the source.

However, this incorporation of peers into the distribution process also introduces new threats on its dependability. In particular, peers are usually unreliable, possess few resources, and are easy to attack. The evolving problems are further intensified by the strict timing requirements caused by the demand on live multimedia data to be available virtually without delay.

By making it possible to reach large audiences despite employing minor resources, peer-to-peer live streaming systems can both support the freedom of information and drive business critical applications. Hence, it is of great importance to ensure lower bounds on the *stability* of these systems. Here, the vague term stability is interpreted as a measure for the ability to withstand different types of destructive events.

In the study of peer-to-peer live streaming systems, the most prominent of such events is a *failure*, i.e., the unexpected and uncoordinated leaving of a set of peers. It is considered in virtually every stability evaluation of newly introduced peer-to-peer streaming systems and is especially studied in [TJ07, VS10, DF10, LCC+11].

In addition to failures, it is also necessary to account for systematic and deliberate denial-of-service *attacks*. These maximize damage by removing a carefully chosen set of peers from the system. Their study is especially important, since considerations of attack-stability have as yet played rather a minor role in the design of peer-to-peer live streaming systems. In particular, publications in this area mostly concentrate on very specific attacks initiated by participating peers, e.g., [WXZJ06, DFK06, DHRS07, GCM11].

A factor that directly influences both failure- and attack-stability of a peer-to-peer streaming system is its *distribution topology*. This graph structure models the distribution process of the stream's packets between the source and the peers. Despite the fact that the stability of peer-to-peer streaming systems is an active field of research, the *analytical* study of the influence of a system's distribution topology on its stability is yet often neglected. In particular, if the distribution topology is actually considered,

authors rely on rules of thumb as building 'short' and 'diverse' trees (e.g., [PWC03]).

In this thesis, we will perform an in-depth study of the interactions between the distribution topology of a peer-to-peer live streaming system and its stability properties. In doing this, we will concentrate on push-based peer-to-peer live streaming systems. They offer the short delivery times and low overhead in the packet distribution process that are necessary for the distribution of live multimedia content. Furthermore, these systems can actively control their distribution topology and hence *optimize it for various stability goals*.

Having a special focus on attack-stability, but also considering failure-stability, we will analytically study the optimization problems posed when aiming to form stable distribution topologies or when trying to find resource-efficient attacks on peer-to-peer streaming systems. In the process, we will identify necessary and sufficient conditions on distribution topologies to achieve the aspired stability goals, point out interesting connections with Graph, Design, and Coding Theory, as well as study the computational complexity of the occuring problems.

1.2. Contributions of this Thesis

The contributions of this thesis can be classified into the following areas:

- We adopt the graph model of [BSS09] for multitree distribution topologies in push-based peer-to-peer live streaming systems. Building on it, we introduce three increasingly complex measures of damage created by attacks on distribution topologies. The LiSS-*damage measure* counts the system-wide number of disturbed source-to-peer paths. It gives a global notion of damage to the streaming service. In contrast, the LoSS-*damage measure* quantifies the number of nodes receiving less than a given fraction of the stream packets. It is especially relevant in situations where Multiple Description Coding is applied to the stream. Finally, the FEC-LoSS-*damage measure* evaluates the number of nodes that are not able to reconstruct the stream data although the stream is encoded using a Forward Error Correction code.

  Based on these measures, we define corresponding optimization problems that aim at creating a desired amount of damage while using a minimum number of removed peers. We identify computational limits of practical attackers by analyzing the complexity and approximability of these problems. In particular, we show that if $\mathbf{P} \neq \mathbf{NP}$, a polynomial-time attacker may have to attack a number of nodes exceeding the possible minimum by a factor growing with the number of distribution trees in the topology. The identified factors and the availability of corresponding planning algorithms depend on the regarded damage function. These results demonstrate the influence of system parameters like tree number and chosen stream encoding on the hardness of planning optimal attacks on peer-to-peer live streaming systems.

- We study distribution topologies that, for every number x of their n peers, minimize the maximum LiSS-damage created by *attacks* removing exactly x

peers. These topologies are called *optimally* LiSS-*stable*.

In particular, we first review the results of [BSS09], giving a characterization of optimally LiSS-stable topologies and a first set of efficiently-identifiable topologies meeting this characterization.

Then, we determine necessary conditions on optimally LiSS-stable topologies and use them to define a small set of rules that describe a much larger and less restrictive set of such stable topologies.

One of these rules imposes the requirement that the supply relationships between the direct neighbors of the source themselves correspond to an optimally LiSS-stable topology. Due to this reason, we specifically investigate these so-called *head topologies*. For this, we state a graph-based version of the characterization of optimally LiSS-stable topologies and again identify (polynomial-time checkable) necessary and sufficient conditions on these topologies.

Finally, we present a result of [Bri08], showing that, although it is possible to efficiently form optimally LiSS-stable topologies with given parameters, it is **coNP**-complete to decide whether a given topology is indeed optimally LiSS-stable. Consequently, if $\mathbf{P} \neq \mathbf{NP}$, we cannot hope to identify arbitrary optimally LiSS-stable topologies in polynomial time.

- We study distribution topologies that, for every number x of their n peers, minimize the maximum LoSS-damage created by *attacks* removing exactly x peers. These topologies are called *optimally* LoSS-*stable*.

  Again, we find necessary conditions on such topologies and identify a dominating part in the studied damage function. We treat this part as a damage measure of its own: the *forward-damage*. Then, we focus on the study of topologies minimizing forward-damage.

  We show that there is a matrix representation of successor relationships in these topologies that characterizes their stability. In particular, we find out that such matrices must be Orthogonal Arrays or specific Packing Arrays. Furthermore, there are close connections to Maximum Distance Separable codes. Finally, we show that if it is possible to identify an efficient general constructing mechanism for the studied topologies, this would solve a number of long-standing research problems in Design and Coding Theory.

- We study distribution topologies minimizing the expected number of lost source-to-peer paths when a set of *failing* peers is chosen uniformly at random. These topologies are called *random-failure-stable*. We find sufficient conditions for these topologies and demonstrate that, depending on the bandwidth available to the source and the peers, there are situations in which no random-failure-stable topology exists, albeit it is generally possible to form distribution topologies. Furthermore, we show that, if they exist, it is an **NP**-complete problem to find random-failure-stable multitree distribution topologies.

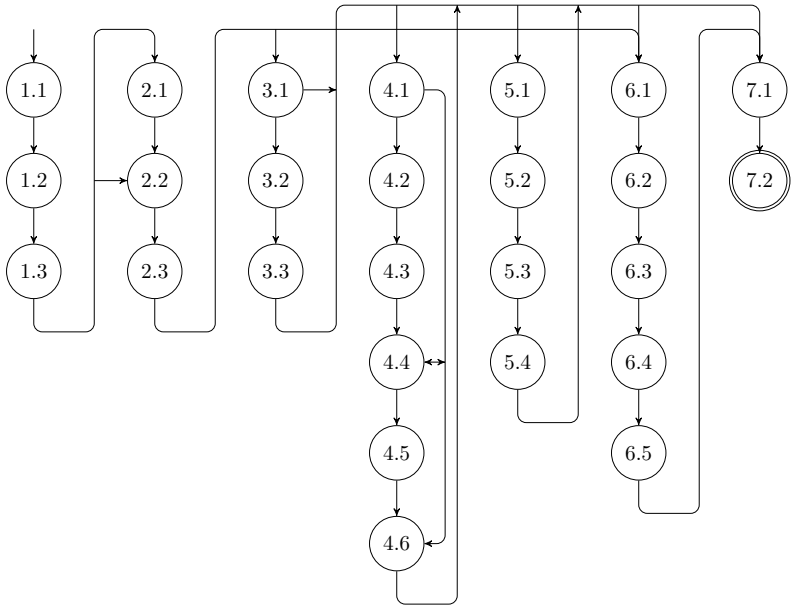


Figure 1.1.: Possible paths through the sections of this document.

1.3. Structure of this Thesis

This document is structured into 7 chapters. Depending on the topical interests and prior knowledge of the reader, it is possible to skip certain topics without impeding the understanding of others. To ease such approaches, Figure 1.1 depicts possible paths through this document and visualizes dependencies between the sections.

Chapter 2 is following this Introduction. It gives a brief background on the technical details of peer-to-peer live streaming systems (Section 2.1), introduces basic building blocks of our mathematical model (Section 2.2), and discusses possible notions of the term *stability* (Section 2.3).

Then, Chapter 3 illustrates our notion of *attacks* on peer-to-peer live streaming systems and introduces the damage functions used to quantify the consequences of such attacks (Section 3.1). This is the fundament for the Chapters 4 and 5, which are also concerned with topics of attack-stability. In the second part of Chapter 3 (beginning with Section 3.2), we study the computational complexity and approximability of problems posed to well-informed but resource-limited attackers on peer-to-peer live streaming systems.

The following Chapters 4, 5, and 6 each focus on finding distribution topologies optimizing a specific stability aspect.

In particular, Chapter 4 covers the optimally LiSS-stable topologies. Its topics include a review of their characterization (Section 4.1), the identification of rules defining a new, large subclass of these topologies for which membership can be checked

in polynomial-time (Section 4.2), a special examination of the important subclass of optimally LiSS-stable head topologies (Section 4.3), the question of the computational complexity of identifying optimally LiSS-stable topologies (Section 4.4), and a brief sketch of heuristics to establish such topologies by way of a distributed topology management (Section 4.5).

Chapter 5 is concerned with optimally LoSS-stable topologies. After identifying basic properties of such topologies (Section 5.1), we show that the optimized damage function is dominated by a specific part which we call *forward-damage* (Section 5.2). The rest of the chapter then focusses on distribution topologies minimizing this forward-damage (Section 5.3).

Chapter 6 studies random-failure-stable distribution topologies. The corresponding extension of our basic model is introduced (Section 6.1) and sufficient conditions for random-failure-stable topologies are identified for single- and multitree topologies (Sections 6.2 and 6.3). Finally, we investigate the computational complexity of finding of random-failure-stable distribution topologies (Section 6.4).

The results and open questions of these topology-centered chapters are summarized in the Sections 4.6, 5.4, and 6.5, respectively.

Chapter 7 concludes this thesis. Section 7.1 gives an overview of the studied problems and obtained results. Finally, Section 7.2 lists open problems and sketches possible future directions of research.

2. Background & Fundamentals

This Chapter introduces the fundamental concepts studied in this thesis.

In particular, Section 2.1 reviews the technical background of peer-to-peer live streaming systems. Thereby, it sketches their importance and motivates the models and problems defined in the subsequent chapters. Section 2.2 establishes our basic model of peer-to-peer streaming distribution topologies. It fixes notations, provides means to describe topologies, and gives a categorization into certain topology classes. Finally, Section 2.3 specifies our general approach to cope with the ambiguity of the term stability in the context of peer-to-peer live streaming systems.

2.1. Peer-to-Peer Live Streaming Systems

This section introduces the basic ideas of peer-to-peer live streaming systems.

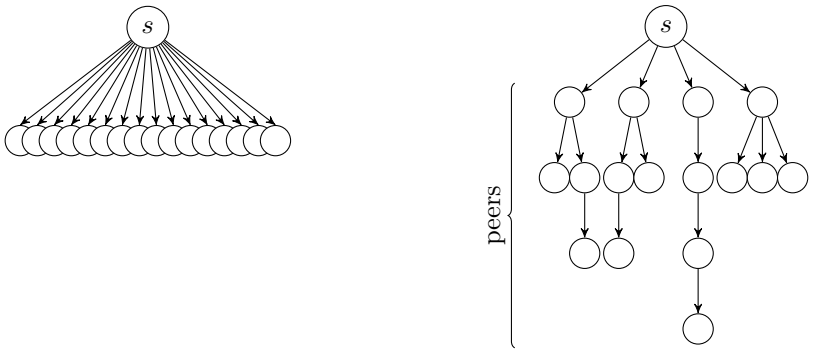
Subsection 2.1.1 describes their purpose and motivation. Furthermore, the most basic building blocks of such a system are presented. The following Subsection 2.1.2 reviews possible encodings of the distributed data stream. These techniques permit to increase the stability of peer-to-peer live streaming and their application is part of our mathematical models. Subsection 2.1.3 introduces the concepts of distribution trees and topologies. Furthermore, the different approaches to create and manage these structures are investigated. Finally, Subsection 2.1.4 points out topology-independent aspects influencing the stability of peer-to-peer live streaming systems.

2.1.1. Purpose and Components

In the recent years, the distribution of realtime-generated multimedia content over the Internet has received constantly increasing popularity. Consequently, by now, the Internet is regarded as a natural distribution channel for live press coverage of important events (e.g., [Aka]), both by content providers and end-users. Furthermore, a wide range of standard television channels is available directly from the channel's website or on IPTV platforms as [PPt] or [Zat].

However, the adoption of this new and low-priced distribution channel introduces a number of fundamental technical problems. One of the main issues is caused by the characteristic traffic pattern of such a multicast, where a large number of participants retrieve the constantly-changing and timing-sensitive data stream from a single *source* for a prolonged period of time.

Consequently, each implementation based on a classical unicast client-server approach has to suffer from a severe scalability problem. Both computational and bandwidth resources of such a server are inherently limited (and expensive).



(a) Client-server distribution pattern. (b) Peer-to-peer distribution pattern.

Figure 2.1.: Examples of client-server and peer-to-peer distribution patterns.

A suitable solution for this problem would lie in the use of IP Multicast [Dee92]. With this standardized technique, the source sends only one copy of its data stream, which is then forwarded and replicated at network layer. For this, a multicast tree is formed whose inner nodes are cooperating Internet routers. However, the decentralized organization of the Internet, the limited number of available multicast group addresses, and open questions about authentication and billing have prevented a wide-spread adoption of this technique. Consequently, it must still be regarded as generally unavailable.

A ready-to-use multicast solution is offered by live streaming systems based on the peer-to-peer approach. Originally used mainly for file-sharing applications, this approach advocates the active participation of clients in the distribution process (cmp. Figure 2.1(b)). These participants, named *peers*, profit from the system and, in exchange, contribute their computational and bandwidth resources. In the particular case of peer-to-peer live streaming systems, the source makes the stream data available to a subset of peers, which then recursively initiate a redistribution among the whole set of peers. Consequently, such a system is able to overcome the scaling limitations of the source, given that the peers possess a certain minimum amount of resources.

The actual transfer of data is based only on pairwise unicast communication.

Not to add another scalability bottleneck, such peer-to-peer systems usually exhibit a decentralized organization without a participant possessing information about the complete system state.

However, to allow new peers to join the running system, it is necessary to supply them with contact information about existing peers. While for small numbers of peers this task is usually carried out by the source, larger systems rely on specialized *bootstrapping servers*. These maintain the addresses of a sample of active peers and provide them on request.

2.1.2. Stream Encoding

Since the Internet is a packet-based network, the continuous multimedia data stream has to be split into blocks (sometimes called chunks) of fixed size. It is then possible to distribute these blocks to the peers, which can reassemble the original data stream as soon as they receive all blocks.

However, the Internet and peer-to-peer systems are unreliable distribution channels, so that losses may occur on the transfer. While these could be handled by requesting a retransmission, the arising delays are hard to reconcile with the strict timing requirements of live media streams. Due to this reason, the stream data blocks are usually *encoded* to mitigate the consequences of such losses.

A frequently proposed technique is *Multiple Description Coding* (MDC) [Goy01]. Here, each stream data block is encoded into $k \in \mathbb{N}$ subblocks, so that the reception of any combination of these subblocks still suffices to play back the stream. In particular, the user-perceived quality gracefully degrades with the number of lost subblocks. This way, a certain amount of losses becomes tolerable, while the redundancy in the encoded stream is very small. The latter is especially important, since the overall bandwidth resources are a limiting factor of peer-to-peer systems, too.

As an alternative, *Forward Error Correcting* (FEC) codes [MS93, LMS+97] can be applied to the stream blocks. Here, a block is encoded into $k \in \mathbb{N}$ subblocks and there is $z \leq k$, such that the *original* block can be reconstructed from any combination of at least z subblocks. However, the complete stream data is lost if less than z subblocks arrive at a peer. Another drawback of FEC codes is that, due to the necessary redundancy, the encoded stream has a considerably higher bitrate than the original stream data. Nonetheless, FEC codes are frequently applied, since the ability to reconstruct the complete original stream blocks is especially useful in the multi-layered distribution process of peer-to-peer live streaming system. With its help, peers can redestribute data which would otherwise have been lost due to problems at their predecessors in the system.

As a third option, a peer-to-peer streaming system can apply *Network Coding* techniques [OSV09]. Again, a stream block is split into k subblocks, but participants can decide to distribute (e.g., linear) combinations of these blocks. Such a dynamic encoding can be used for forward error correction, but also to overcome network bottlenecks. However, to profit from such an approach, a peer-to-peer system must either give its peers great liberty in their choice of forwarded data and receiving peers, or contain altruistic peers which are not interested in decoding the stream themselves. Although there are peer-to-peer live streaming systems meeting these conditions (e.g., [GZL03, WL07]), the push-based multitree systems (to be introduced in Section 2.1.3) on which we will focus in this thesis do not. For this reason, Network Coding techniques will not be further considered.

2.1.3. Distribution Trees and Topology Management

Assuming a static, source-based encoding of the stream, each new packet that the source injects into the peer-to-peer live streaming system is forwarded to all participating

peers. In particular, each peer seeks to receive the packet exactly once. Consequently, when forming a graph with all participants as nodes and the pairwise connections crossed by the replications of such a packet as edges, we obtain a *distribution tree*. This tree is rooted at the source and its edges are directed towards its leafs.

There are two basic competing concepts to organize such distribution trees in a peer-to-peer live streaming system.

The *mesh-based* (also named *pull-based*) approach, as adopted by [ZLLY05, PPt, MR07] and many more, is technically very similar to peer-to-peer file sharing systems. The source and the peers are arranged in a connected graph, called the overlay. In this graph, each participant has a relatively small number of neighbors. Additionally, the participants have local memory of constant size, named buffer. It contains the newest stream packets they are possessing and neighbors regularly exchange descriptions of their available packets. This way, a peer can identify lacking packets in its neighbors' buffer and actively requests their forwarding.

This easy-to-implement approach guarantees a continuous adaption to overlay changes and can theoretically result in a large number of different distribution trees for the sent packets. However, measurements [WLX08] show that the variance between the actually used distribution trees is quite limited. This fact motivated systems like [MR07], trying to increase this variance by letting the source partition its neighbor set and enforcing that different packets are forwarded to these neighbor subsets.

Although wide-spread in practical applications, mesh-based peer-to-peer live streaming systems have severe drawbacks, since the interactive distribution protocol leads to high forwarding overhead, unpredictable packet delivery times, and long delays. Each of these points critically contrains their usefulness for the distribution of timing-sensitive live multimedia data.

In contrast, fast and predictable forwarding with minor control overhead are features of the *tree-based* (also named *push-based*) approach. Here, peers arrange in static tree structures that are maintained for the whole time of system operation. The stream packets are then sent along these predefined trees. In particular, upon the reception of a new packet, a peer immediately starts to retransmit the data to its downstream neighbors in the same tree.

While the initially proposed systems [GJKJ00, CDKR02, CRSZ02, BBK02, THD03, Cha03, TJD+05] relied only on a single static distribution tree, by now the use of multiple distribution trees is predominant [PWCS02, CDK+03, VYF06, BSS09]. Systems following this approach are also called *multitree* systems. They offer a higher number of paths between the source and each peer, which dramatically increases the system's stability when the different trees are used to distribute the subblocks of an MDC or FEC encoded stream block. Furthermore, a higher number of trees allows to split the stream data into smaller portions, called stripes. Since the minimum upload bandwidth necessary to forward a stripe is only a fraction of the stream bandwidth, it becomes easier to integrate peers with asymmetric bandwidth capacities.

The set of distribution trees of a tree-based peer-to-peer live streaming system forms its distribution topology. In this thesis, we will call both a static distribution tree and the share of information transmitted over it a *stripe*.

The downside of the tree-based approach is that the sudden exit of a peer in an

important tree position can lead to considerable down-times in the corresponding subtree due to repairs. Consequently, sophisticated repair mechanisms need to be applied [FY07, BFGS09a]. Generally, tree maintainance costs evolve, which clearly grow with the amount of change in the peer set. However, the involved topology awareness can be further utilized.

A distribution topology reflects the individual relationships of packet reception and redistribution between the stream source and the participating peers. It therefore models the core mechanisms determining the service parameters of the streaming system, including – as we will see later – its vulnerability against different kinds of failures and attacks.

Due to this fact, push-based peer-to-peer live streaming systems should actively improve their distribution topology over time. During this process, a streaming system will aim to build a topology that is *optimal* considering one or several optimization goals. One of the central contributions of this thesis lies in the identification and study of classes of topologies, that are – in a specific sense – *optimally stable*. The actual notions of *stability* that are considered, are motivated by attacks on and failures in real-world streaming systems. They will be legitimated in Section 2.3.

Proposed mechanisms for a distributed topology management include global schemes assigning peer positions based on random peer-identifiers [CDK+03], incremental optimization of peer-local cost functions [Str07, BSS09], and the application of Network Motifs [KAS10].

Recently, there is an increasing number of *hybrid* approaches, trying to combine the advantages of both mesh- and tree-based systems. Such a hybrid system can either start with a mesh-based approach and then switch to automatic forwarding on long-term relationships between peers (e.g., [LMSW07, WLX08, WXL10, WLX11]) or it forms an explicit static distribution topology that is combined with mesh-based backup mechanisms in the case of failures (e.g., [YLY+04, ZZSY07, PPK10]). Since both approaches form static distribution trees spanning over either the long-living or all peers, the results of this thesis are equally relevant for this emerging category of peer-to-peer live streaming systems.

2.1.4. Further Aspects of Peer-to-Peer Live Streaming System Stability

Although this thesis focuses on the connections between distribution topologies and system stability, there are a number of further measures improving the stability of peer-to-peer live streaming systems. Many of them aim at securing the interactions of peers that lead to the formation of the distribution topology. In that, they are more specific to certain system implementations.

In [BFGS09a], the author of this thesis participated in an investigation of the design of topology management mechanisms that are resistant to malicious manipulations initiated by peers. Necessary features include a cautious use of unconfirmed performance data obtained from single peers, a source-directed flow of topology information, peer-selection strategies based on long-term information, and proactive preparations for the loss of important peers (e.g., see also [FY07]).

In this context, especially the peer selection strategies are an active field of research. In [TWSN08, VS10, KSU11] peer lifetime models are stated that aim at predicting the probability of an approaching peer exit. Other solutions assign a global trust or reputation value to each peer, growing with its service to the streaming system [WXZJ06, RSS07]. Due to the shorter pairwise relationships, this is especially important when a mesh-based approach is adopted [HvR08, SSNRR10]. If such reputation systems are combined with admission control mechanisms, it is also possible to ban malicious peers from the system [XZ06, ZYL+07].

A connected topic is the introduction of incentives, motivating selfish peers to contribute more bandwidth resources to the streaming system (e.g., [SFC08, PPK10]). Such mechanisms indirectly improve system stability by providing higher flexibility for the topology construction and decreasing the dependence on single bandwidth-contributors. As a complementing approach, it was also proposed to dynamically adapt the source's bandwidth capacity [FKGS11, SIB12] by ways of virtualization or cloud support.

Since none of these approaches directly influences the actually formed distribution topology, they could be smoothly integrated in a peer-to-peer live streaming system that aims at building stable distribution topologies as identified in this thesis.

2.2. A Model For Distribution Topologies of P2P Live Streaming Systems

The tool enabling all our analysis and results is a graph-based mathematical model of peer-to-peer distribution topologies. Its elementary ideas first occured in [BSS09] and were later extended and adapted by the author. Note that this section restricts to the very basic parts of our model and that the Chapters 3–6 will add additional components where necessary. This approach will support readers who are interested in only specific aspects of topology stability and keeps definitions local.

2.2.1. Notations and Specifications

Before introducing the actual model, let us fix some notations and concepts of basic mathematical objects. To avoid common ambiguities, we define the set of *natural numbers*

$$\mathbb{N} := \{i \in \mathbb{Z} \mid i \geq 1\} \tag{2.1}$$

as the set of integers of value *at least* one. Furthermore, given $a, b \in \mathbb{Z}$, we define the integer interval with borders a and b as

$$[a, b] := \{i \in \mathbb{Z} \mid a \leq i \leq b\} \tag{2.2}$$

and write

$$[a] := [1, a]. \tag{2.3}$$

We assume basic knowledge of the concepts of Graph Theory. A thorough introduction

on this topic can be found in [Die05]. If the meaning is evident from the context, we will use the word graph for both simple graphs and multigraphs.

For a multigraph $G = (V, E)$ and $x, y \in V$, the *multiplicity* $m_G(x, y)$ denotes the number of parallel edges $\{x, y\}$ in E. The multiplicity of two node sets $X, Y \subseteq V$ is

$$m_G(X, Y) := \sum_{x \in X} \sum_{y \in Y \setminus X} m_G(x, y). \tag{2.4}$$

Finally, the *multiplicity of a single node* $v \in V$ is $m_G(v) := m_G(\{v\}, V)$. In a simple graph, it coincides with the degree of v. A multigraph $G = (V, E)$ is called *r-regular*, if it holds that $\forall v \in V : m_G(v) = r$.

A multigraph $G = (V, E)$ is a *clique*, if $\forall u, v \in V : u = v \vee m_G(u, v) \geq 1$. In case of equality, it is called a *simple clique*.

Given $G = (V, E)$ and $X \subseteq V$, we denote the submultigraph of G induced by X as $G[X]$. Furthermore, the number of edges in $G[X]$ is denoted as

$$e_G(X) := \frac{1}{2} \sum_{v \in X} m_{G[X]}(v). \tag{2.5}$$

When speaking about properties of algorithms and computational problems, the word *complexity* is usually referring to the concept of time complexity [Weg05]. Furthermore, an algorithm is called *efficient*, if it runs in a time that is polynomial in the length of the input in a binary representation. If considering computational problems for which the length of a solution is at least polynomial in the highest numeric value of the input (i.e., it is *pseudopolynomial*), we also regard pseudopolynomial-time algorithms as efficient.

2.2.2. Basic Topology Model

Following the discussion in Section 2.1.3, the distribution topology of a push-based peer-to-peer streaming system consists of one or multiple trees that are rooted at the source node and span over all participating peers. In this thesis, we assume that the stream data is split into $k \in \mathbb{N}$ substreams called *stripes*. Each stripe is distributed over a corresponding distribution tree. Such a tree is also called *stripe tree* or just *stripe*.

Definition 2.2.1 ***Distribution Topology*** $\mathcal{T}$
Given $k \in \mathbb{N}$ and a node set V, a *distribution topology $\mathcal{T}$ of k stripes over node set V* is a k-tuple $\mathcal{T} := (T_1, \ldots, T_k)$ of directed trees, each rooted at the same distinguished node $s \notin V$, containing exactly the nodes $\{s\} \cup V$, and with edges that are directed towards the tree leafs. A tree T_i from $\mathcal{T}$ is also called *stripe i*. The nodes V are called *peers*.

Abusing notation, we will sometimes interpret a topology $\mathcal{T}$ as a multiset of stripe trees and write $T \in \mathcal{T}$ to denote that T is a tree of $\mathcal{T}$.

To obtain a model that is both compact and meaningful, we make a number of abstractions from real-world streaming systems:

- *Idealized Underlay Topology*: In applied peer-to-peer live streaming systems, peers correspond to nodes in an underlying communication network, e.g., the Internet. A peer u can communicate with a peer v if there is a $u \to v$ path in this network. Hence, for peer-to-peer streaming to be possible, we can assume that this network is at least weakly connected and that there is a path from the source to each participating peer.

  However, usually the network is *much* more densly connected, as today's communication network protocols rely on bidirectional communication. Especially when considering the IP-based Internet, the graph of pairwise reachability of nodes contains a large clique. Here, the biggest problem in reachability is posed by the existence of subnetworks using Network-Address-Translation (NAT), thereby allowing bidirectional connections only when they are initiated from a node inside the subnetwork. However, there are a number of techniques (e.g., [FSK05], [WSHW08]), helping peer-to-peer systems to initiate connections also in the presence of NAT gateways.

  Due to these reasons, in this thesis, we will generally assume complete pairwise reachability of all participating nodes. Furthermore, we will not include the underlying communication network in our model. However, note that the author also contributed to [FKGS11, FDGS11], where topics of overlay/underlay-interactions were considered.

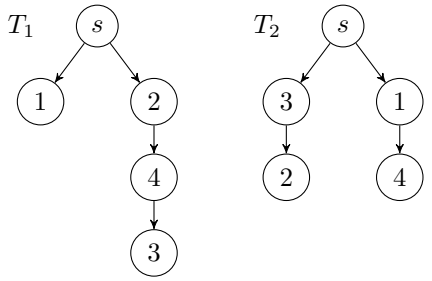
- *Static System Parameters*: Real-world peer-to-peer streaming systems are dynamic systems with a constantly changing set of peers. Therefore, the topologies of real-world peer-to-peer streaming systems are subject to change, too. However, between each two consequtive changes, the system is in a static state. For each such state of static system parameters, the topology management aims at constructing a topology that is in some sense optimal. Hence, to gain optimality for the dynamic system, we can break these dynamics down into a sequence of static states and study what characterizes optimal topologies when the system parameters are unchanged. Due to this reason, in this thesis, we fully concentrate on finding optimal topologies in situations without changing topology parameters and without peer dynamics.

We now introduce our tools to describe distribution topologies.

Depth, Depth Levels, and Heads Let $\mathcal{T}$ be a distribution topology with k stripes on node set V and let $T \in \mathcal{T}$. The *depth* $d_T(v)$ of a node $v \in V$ in tree T is specified by the number of edges on the unique $s \to v$ path in T. The *depth of tree* $T = (\{s\} \cup V, E)$ is defined as $d(T) = \max_{v \in \{s\} \cup V} d_T(v)$ and the *depth of topology* $\mathcal{T}$ is $d(\mathcal{T}) = \max_{T \in \mathcal{T}} d(T)$.

For $i \in [0, |V|]$, the *depth level* i *of tree* T is

$$L_i(T) := \{v \in V \mid d_T(v) = i\} \tag{2.6}$$



(a) Topology $\mathcal{T}$

Property	Value
$d(\mathcal{T})$	3
$H_1^{\mathcal{T}}$	$\{1, 2\}$
$H^{\mathcal{T}}$	$\{1, 2, 3\}$
$L_2(\mathcal{T})$	$\{(4, 1), (2, 2), (4, 2)\}$
$\text{pred}_2^{\mathcal{T}}(4)$	$\{s, 1, 4\}$
$\text{succ}_2^{\mathcal{T}}(3)$	$\{2, 3\}$
$\text{succ}_2^{\mathcal{T}\rightarrow}(3)$	$\{2, 3\}$
$\text{succ}_1^{\mathcal{T}\rightarrow}(3)$	$\emptyset$

(b) Selected Properties of $\mathcal{T}$.

Figure 2.2.: A distribution topology $\mathcal{T} = (T_1, T_2)$ and some of its properties.

and the *depth level i of topology $\mathcal{T}$* is the *disjoint union*

$$L_i(\mathcal{T}) := \bigsqcup_{T \in \mathcal{T}} L_i(T), \tag{2.7}$$

e.g., it holds that $|L_i(\mathcal{T})| = \sum_{T \in \mathcal{T}} |L_i(T)|$.

We will see, that the nodes of depth one play a very important role for the stability properties of $\mathcal{T}$. Therefore, we will call these nodes the *heads of topology $\mathcal{T}$*. For each stripe $T_i \in \mathcal{T}$, we define the set

$$H_i^{\mathcal{T}} := L_1(T_i) \tag{2.8}$$

and call

$$H^{\mathcal{T}} := \bigcup_{T_i \in \mathcal{T}} H_i^{\mathcal{T}} \tag{2.9}$$

the *head set of $\mathcal{T}$*. Note that $H^{\mathcal{T}}$ is *not* a disjoint union.

Predecessors and Successors The nodes V can be of different relevance to the topology $\mathcal{T}$. Although depth already gave us a notion of such relevance, it is better characterized by predecessor and successor relationships of nodes.

Definition 2.2.2 ***Predecessor Sets***
Let $\mathcal{T} = (T_1, \ldots, T_k)$ be a distribution topology on node set V and let $v \in V$.
The *predecessor set of v in stripe i of $\mathcal{T}$* is defined as

$$\text{pred}_i^{\mathcal{T}}(v) := \{s, u_1, \ldots, u_l, v\},$$

such that $(s, u_1, \ldots, u_l, v)$ is the node order on the $s \rightarrow v$ path in T_i.

Definition 2.2.3 ***Successor Sets***
Let $\mathcal{T} = (T_1, \ldots, T_k)$ be a distribution topology on node set V and let $v \in V$.
The *successor set of v in stripe i of $\mathcal{T}$* is defined as

$$\operatorname{succ}_i^{\mathcal{T}}(v) := \{u \in V \mid v \in \operatorname{pred}_i^{\mathcal{T}}(u)\}.$$

Furthermore, for $X \subseteq V$, we define

$$\operatorname{succ}_i^{\mathcal{T}}(X) := \bigcup_{v \in X} \operatorname{succ}_i^{\mathcal{T}}(v).$$

Note that these definitions imply $\forall i \in [k]\colon v \in \operatorname{pred}_i^{\mathcal{T}}(v) \cap \operatorname{succ}_i^{\mathcal{T}}(v)$.
A special subset of $\operatorname{succ}_i^{\mathcal{T}}(v)$ is the set of v's children

$$\operatorname{child}_i^{\mathcal{T}}(v) := \{u \in \operatorname{succ}_i^{\mathcal{T}}(v) \mid (v, u) \text{ is edge in } T_i\}. \tag{2.10}$$

Additionally, we define the one-element-set

$$\operatorname{parent}_i^{\mathcal{T}}(v) := \{u \in \operatorname{pred}_i^{\mathcal{T}}(v) \mid v \in \operatorname{child}_i^{\mathcal{T}}(u)\}. \tag{2.11}$$

The above sets are sufficient to give a complete formal description of the topology $\mathcal{T}$. However, the following, slightly different notion of successor sets will be useful in the Chapters 4 and 5.

Definition 2.2.4 ***Forward Successor Sets***
For a distribution topology $\mathcal{T} = (T_1, \ldots, T_k)$ on node set V and $v \in V$, define the *forward successor set of v in stripe i of $\mathcal{T}$* as

$$\operatorname{succ}_i^{\mathcal{T}\rightarrow}(v) := \begin{cases} \operatorname{succ}_i^{\mathcal{T}}(v) & , \text{ if } |\operatorname{succ}_i^{\mathcal{T}}(v)| > 1 \vee v \in H_i^{\mathcal{T}} \\ \emptyset & , \text{ otherwise.} \end{cases}$$

Furthermore, for $X \subseteq V$, we define $\operatorname{succ}_i^{\mathcal{T}\rightarrow}(X) := \bigcup_{v \in X} \operatorname{succ}_i^{\mathcal{T}\rightarrow}(v)$.

In contrast to the successor set of node v in stripe i, the forward successor set is empty if v neither forwards nor is head in stripe i. By specially treating heads, we guarantee that $\forall i \in [k]\colon \operatorname{succ}_i^{\mathcal{T}\rightarrow}(H_i^{\mathcal{T}}) = V$.
Figure 2.2 recapitulates the above definitions in an example.

2.2.3. Classes of Distribution Topologies

When studying distribution topologies, it will be useful to specify sets of distribution topologies with similar parameters. In the context of this thesis, we will denote such sets as *classes of distribution topologies*.

Definition 2.2.5 ***Bandwidth-Restricted Distribution Topologies***
Let $n, k \in \mathbb{N}$ be given, let $V = [n]$ be a set of nodes and let s be a distinguished source node with $s \notin V$. For a capacity function $\mathrm{c}\colon \{s\} \cup V \to \mathbb{N} \cup \{\infty\}$ specifying a maximum out-degree per node, the class of *Bandwidth-Restricted Distribution Topologies* $\mathbb{T}(n, \mathrm{c}, k)$ is defined as the set of distribution topologies with k stripes, source node s, node set V, and the property

$$\forall v \in \{s\} \cup V\colon \sum_{i \in [k]} |\mathrm{child}_i^{\mathcal{T}}(v)| \leq \mathrm{c}(v).$$

In this definition we implicitely assume that all stripes of a stream have equal bandwidth demands. If the streaming system is distributing data stemming from an MDC- or FEC-encoded multimedia stream (cmp. Section 2.1.2), this is a common abstraction.

Since a distribution topology is a set of k trees over $V \cup \{s\}$ rooted at s, such a class will be empty if $\mathrm{c}(s) < k$ (less than k stripes can be rooted at s) or $\mathrm{c}(s) + \sum_{v \in V} \mathrm{c}(v) < kn$ (k trees on $n+1$ nodes need kn edges).

In many cases, we will not impose limiting bandwidth assumptions for *peer nodes*, since we will be interested in topology properties that are independent of these limitations. Clearly, a class without bandwidth restrictions for peer nodes will also contain all topologies that would have met a certain restriction.

Furthermore, in most parts of this thesis, we will make the assumption that the source node is able to transmit the whole stream (i.e., all k stripes) exactly C times for a $C \in \mathbb{N}$. Thus, it holds that $\mathrm{c}(s) = Ck$.

Combining these assumptions, we define our standard class $\mathbb{T}(n, C, k)$ of distribution topologies.

Definition 2.2.6 ***The Class*** $\mathbb{T}(n, C, k)$
For $C, k \in \mathbb{N}$ and $n \geq Ck$ we define

$$\mathbb{T}(n, C, k) := \mathbb{T}(n, \mathrm{c}, k)$$

with

$$\mathrm{c}(v) := \begin{cases} Ck & \text{, if } v = s, \\ \infty & \text{, otherwise.} \end{cases}$$

The Parameters n, C and k in Real-World Applications In practical peer-to-peer streaming systems, we can assume that $n \gg Ck$, since measurements [SMZ04, WLZ08, WLX08] indicate that popular daily streams attract multiple tens of thousands of concurrent users and that the general demand for live multimedia streaming is rapidly growing. In particular, the server-based Content Distribution Network Akamai announced that it delivered live streams of the Soccer World Cup 2010 and Great Britain's 2011 royal wedding, both, to over 1.6 million concurrent clients [Aka].

More interesting are the parameters C and k, which can be directly influenced by the operator of the streaming system. Here, C and k correspond to very different aspects

of the system. While C is determined by the bandwidth and computational resources of the source node (or the monetary resources of the organization paying for both), the parameter k is essentially limited by the bitrate of the distributed stream and subject to a difficult trade-off.

On the one hand, splitting the stream into a high number of stripes minimizes the bandwidth burdens for peers distributing a stripe. Thus, it involves even nodes with minor upload bandwidth and allows for a high number of children. Therefore, the individual stripe trees can be built with small depth. Furthermore, as we will see in the Chapters 3–5, a high parameter k can translate into an increase of topology stability towards deliberate attacks.

On the other hand, the granularity of the applied stream encoding techniques (cmp. Section 2.1.2) is limited. Furthermore, the maintainance of a higher number of stripe trees not only demands for additional computing resources, but also constantly worsens the proportion of actually transmitted streaming data to the necessary overhead of control data that has to be sent.

2.3. Notions of Stability: Measuring an Ambiguous Concept

We should now give a clear specification of one of the most central concepts of this thesis: our exact interpretation of the term *stability*.

When studying the scientific literature, we encounter a multitude of different notions of the stability of peer-to-peer streaming distribution topologies. Terms like stability, resilience, and robustness are brought up and either interpreted as synonyms or beheld as completely different concepts. Frequently, analysis of topology stability is mixed (or confused) with analyses of the behavior and security of the *management mechanisms* of peer-to-peer streaming systems and their actual *implementations* (e.g., cmp. Section 2.1.4). Furthermore, the confusion peaks when it comes to measuring and quantifying each of these concepts.

The reasons for this unsatisfactory state lie in the high complexity and ambiguity of the modeled object itself. Peer-to-peer streaming systems have many different applications, which lead to diverse and possibly conflicting sets of demands.

Consequently, there can be no one-fits-all concept of distribution topology stability. Therefore, we will study *several different notions of stability* in this thesis. However, in doing this, we will always pursue the following approach.

In general, we will interprete distribution topology stability as a property by which distribution topologies can be categorized according to the consequences of certain kinds of disruptive events. The incarnation of such an event is the sudden removal of a set of nodes from the distribution topology. Depending on the circumstances under which such a removal occurs, we will call the event (and the removed node set) either an *attack* or a *failure*. This distinction determines the evaluation of an event's impact:

- An *attack* is a deliberate and malicious attempt to create a maximum degree of non-functionality inside the streaming system, which is measured as *damage*. An

attack is planned and carried out by an entity called the *attacker*. The attacked node set is chosen *based on information about the distribution topology* and its cardinality is limited by the resources of the attacker.

Notions of stability that are based on attacks will generally be called *attack-stability*. We will see that different notions of *damage* (introduced in Section 3.1) lead to very different notions of attack-stability. Nonetheless, they are all based on *worst-case assumptions* about the information and the planning resources of the attacker.

We will study attack-stability from different perspectives. In Section 3.2, we investigate the complexity and approximability of problems aiming to find resource-efficient attacks. In contrast, the Chapters 4 and 5 are devoted to the identification of topologies that minimize the maximum damage that an attack can achieve on them.

- A *failure* corresponds to the unexpected removal of a set of nodes due to coincidental malfunctions, loss of interest, or for other unplanned reasons. In that, a failure is an uncoordinated event. The consequences of a failure can again be measured as damage. However, here an *average-case* perspective is adopted.

  Node failures and their consequences can be modeled by a random process. Topology stability concerning such random failures will be the topic of Chapter 6.

3. Attacks, Damage Measures, and Attack Problems

Following the discussion in Section 2.3, in this chapter we start our analysis of the attack-stability of peer-to-peer live streaming distribution topologies.

For this, Section 3.1 first presents an abstract model of attacks on these systems. Distribution topologies are used to measure the negative effects of such attacks. Their consequences are studied with both a global view on the system and a local view on the perceived streaming quality of individual peers. The definitions of this section also provide the basis for our analysis in the Chapters 4 and 5.

The second part of this chapter is dedicated to the study of attack problems on distribution topologies. In particular, we investigate the complexity and approximability of computational problems posed to a well-informed attacker with limited attack-resources. This is the topic of Section 3.2. Such an approach contributes to our study of attack-stability, since it allows to identify factors complicating the planning of resource-efficient attacks. The results of this chapter are summarized in Section 3.3.

3.1. Quantifying the Consequences of an Attack

In real-world peer-to-peer live streaming systems, there are numerous different possibilities to attack and disturb the functionality of stream distribution. Surveys [Fis12, GCM11] distinguish between attacks initiated by peers, external parties (e.g., Botnets) or a combination of both. The medium of attack can be resource-exhaustion, exploitation of incorrect implementations, a coordinated stop of stream forwarding, tampering with stream data (which both have equal consequences if the stripes are authenticated by cryptographic means), Sybil- and Eclipse-attacks, and many more.

The demands that such a situation poses on the design of dynamic topology management mechanisms have been studied under participation of the author in [BFGS09a].

However, in this thesis we choose an abstracting approach that unifies the actual effects of all these real-world attacks in a simple but powerful model. In particular, we will interprete an attack simply as a set of nodes that are removed from the topology.

Definition 3.1.1 ***Attack*** X
Given a distribution topology $\mathcal{T}$ on node set V, an *attack on* $\mathcal{T}$ is a set $X \subseteq V$.

Such an attack is chosen by the attacker based on information about the topology $\mathcal{T}$ and is limited in its cardinality by the resources of the attacker.

In general, we will follow the worst-case assumption that the attacker has knowledge of the complete topology $\mathcal{T}$. All hardness results on attack problems and all stability

results for topologies that are obtained under this assumption will equally hold for attackers with only partial knowledge.

Note that we explicitly exclude the source s from being attacked. This specification is necessary, since otherwise any attack containing the source would break the functionality of the complete streaming system, regardless of the system's topology. Besides implying equal stability of peer-to-peer streaming systems and server-based streaming, this would hinder us to study the attack-stability of the distribution topology itself. Furthermore, this assumption is reasonable, since in the practical application of peer-to-peer live streaming systems, the great majority of peer nodes belongs to end-users and will be much more vulnerable to an attack than the explicitly secured source infrastructure.

Given a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ and an attack $X \subseteq V$ on $\mathcal{T}$ (note that V is defined by the class $\mathbb{T}(n, C, k)$), we can now quantify the *damage* that the removal of the nodes X will do to $\mathcal{T}$. We will distinguish three different types of damage, which are all motivated by real-world applications.

The most simple damage measure is the *packet loss* or LiSS-*damage*. The definition used here, first occured in [BSS09].

Definition 3.1.2 LiSS-*Damage / Packet Loss* $a^{\mathcal{T}}(X)$
Given a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ and an attack $X \subseteq V$ on $\mathcal{T}$, the *packet loss* or LiSS-*damage* of X is defined as

$$a^{\mathcal{T}}(X) := \sum_{i \in [k]} |\mathrm{succ}_i^{\mathcal{T}}(X)|.$$

Figure 3.1 gives an example. This damage function counts the number of source-to-peer paths in $\mathcal{T}$ that are disturbed by the removal of the nodes X from $\mathcal{T}$. It holds that $0 \leq a^{\mathcal{T}}(X) \leq kn$. Since none of the lost paths can be used for the reception of the next packets sent along the stripes, $a^{\mathcal{T}}(X)$ is sometimes called the *packet loss* measure. Furthermore, the function was, historically, named LiSS-damage due to its close connection with the *Minimum Live Streaming Stability Problem* (LiSS). The latter will be studied in Section 3.2.

The LiSS-damage gives a good *global* impression on topology damage, but it neglects the effects that an attack has on individual peers. Although two attacks X and Y on a topology $\mathcal{T}$ may lead to the same global packet loss, it is possible that the loss of incoming paths is distributed very differently on the peer nodes. For example, every single peer of $\mathcal{T}$ might lose only a few stripes due to attack X, whereas Y might concentrate the same packet loss on a small number of nodes. Depending on the encoding technique of the stream (see Section 2.1.2), these attacks will result in very different receptions of service quality for the individual peer nodes. If Multiple Description Coding is applied, a multimedia stream received in only a subset of stripes continues to be playable and just loses quality depending on the number of lost packets. With Forward-Error-Correction encoding, it is even possible to completely compensate the loss of a certain fraction of packets.

These ideas are reflected in the definition of LoSS- and FEC-LoSS-damage. At first, we determine the number of trees in which a node $v \in V$ can be reached from the

source despite the removal of X:

$$\mathrm{inc}_X^{\mathcal{T}}(v) := |\{T_i \in \mathcal{T} \mid v \notin \mathrm{succ}_i^{\mathcal{T}}(X)\}| \tag{3.1}$$

If the $s \to v$ path in tree T_i is disturbed, we informally say that v *has lost stripe* i. Assuming that $\mathcal{T} \in \mathbb{T}(n, C, k)$ and given a number $z \in [k]$ of stripes, we can then define the *Lost Service Set under Multiple Description Coding* $L_{X,z}^{\mathrm{MDC}}$ as the set of nodes which have lost at least z of their k stripes

$$L_{X,z}^{\mathrm{MDC}} := \{v \in V \mid \mathrm{inc}_X^{\mathcal{T}}(v) \leq k - z\}. \tag{3.2}$$

Hence, we can express a level of tolerance towards stripe loss at individual nodes and declare a node as damaged as soon as it has lost at least z stripes.

Definition 3.1.3 LoSS-*Damage* $\mathrm{b}^{\mathcal{T}}(X, z)$
Given a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, a service loss threshold $z \in [k]$, and an attack $X \subseteq V$, the LoSS-*damage* of X is defined as

$$\mathrm{b}^{\mathcal{T}}(X, z) := |L_{X,z}^{\mathrm{MDC}}|.$$

Again, the name LoSS-damage originates from a connection with the *Minimum Local Streaming Stability Problem* (LoSS) which will be studied in Section 3.2.

Finally, when considering an FEC encoding of the stream, each peer can reconstruct the whole stream data in spite of the loss of up to $z - 1$ of the total k stripes. The value of z is chosen by the operator of the streaming system as a trade-off between error-correction capability and consumption of additional bandwidth resources. With such an encoding, a peer v can redistribute a stripe i albeit some of its predecessors in T_i have been attacked. The precondition for such a reconstruction is that v still receives at least $k - z + 1$ other stripes.

To model this situation, we introduce the predicate

$$\mathrm{avail}_{X,z,N}^{\mathcal{T}}(v, i) := \begin{cases} 1 & \text{, if } v \notin \mathrm{succ}_i^{\mathcal{T}}(X) \\ 1 & \text{, if } v \in \mathrm{succ}_i^{\mathcal{T}}(X) \setminus X \wedge (v, i) \notin N \wedge \\ & \quad \Big(\mathrm{avail}_{X,z,N\cup\{(v,i)\}}^{\mathcal{T}}\big(\mathrm{parent}_i^{\mathcal{T}}(v), i\big) = 1 \vee \\ & \quad\ \sum_{j=1}^{k} \mathrm{avail}_{X,z,N\cup\{(v,i)\}}^{\mathcal{T}}(v, j) > k - z\Big) \\ 0 & \text{, otherwise.} \end{cases} \tag{3.3}$$

Called with $N = \emptyset$, it determines whether stripe i is available at node v assuming attack X and service loss threshold z. If v is not a successor of an attacked node in T_i, the predicate has the value 1. Otherwise, it is also 1 if v is not attacked itself and either stripe i can be obtained from v's parent in T_i or more than $k - z$ stripes are available at v. In the evaluation, the set N is used to break cyclic dependencies. With the help of $\mathrm{avail}_{X,z,N}^{\mathcal{T}}(v, i)$, we define the *Lost Service Set under Forward Error*

Correction $L^{\mathrm{FEC}}_{X,z}$ as

$$L^{\mathrm{FEC}}_{X,z} := \Big\{ v \in V \;\Big|\; \sum_{i=1}^{k} \mathrm{avail}^{\mathcal{T}}_{X,z,\emptyset}(v,i) \leq k - z \Big\}. \tag{3.4}$$

$L^{\mathrm{FEC}}_{X,z}$ contains all the nodes that, after the removal of X from $\mathcal{T}$, are no longer capable to decode the stream. Its cardinality will serve as our third damage measure.

Definition 3.1.4 FEC-LoSS-*Damage* $\mathrm{bec}^{\mathcal{T}}(X,z)$
Given a topology $\mathcal{T} \in \mathbb{T}(n,C,k)$, a service loss threshold $z \in [k]$, and an attack $X \subseteq V$, the FEC-LoSS-*damage* of X is defined as

$$\mathrm{bec}^{\mathcal{T}}(X,z) := |L^{\mathrm{FEC}}_{X,z}|.$$

Both FEC-LoSS- and LoSS-damage count nodes having lost at least z stripes. Since the latter does not account for stripe reconstruction capabilities, we obtain the following relation:

$$0 \leq \mathrm{bec}^{\mathcal{T}}(X,z) \leq \mathrm{b}^{\mathcal{T}}(X,z) \leq n. \tag{3.5}$$

Furthermore, on a topology $\mathcal{T} \in \mathbb{T}(n,C,k)$ of depth at most 2, for every $X \subseteq V$ and every $z \in [k]$, it holds that

$$\mathrm{inc}^{\mathcal{T}}_X(v) \leq k - z \Leftrightarrow v \in X \vee |\{i \in [k] \mid \mathrm{parent}^{\mathcal{T}}_i(v) \in X\}| \geq z \tag{3.6}$$

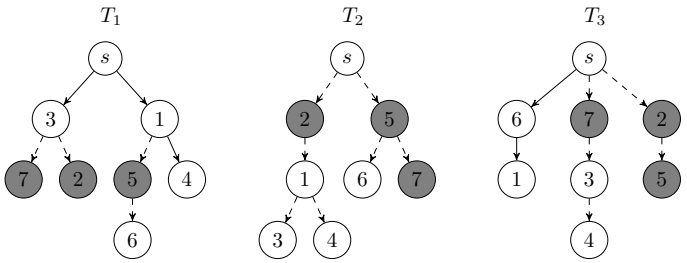
$$\Leftrightarrow \sum_{i=1}^{k} \mathrm{avail}^{\mathcal{T}}_{X,z,\emptyset}(v,i) \leq k - z. \tag{3.7}$$

Consequently, LoSS- and FEC-LoSS-damage are equivalent in this case.

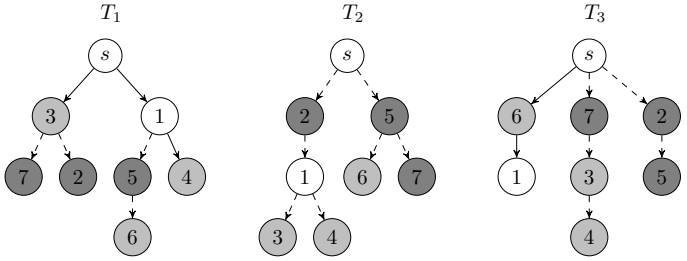
Figure 3.1 recapitulates all defined types of damage in an example.

These formal, yet practically motivated damage functions are the basis for all our further studies on attack-stability. In particular, Section 3.2 will investigate the influence of the damage function on the computational complexity and approximability of a problem imposed to resource-limited attackers: Given a topology and certain parameters of an attack, including a damage threshold, find an attack of minimum cardinality that satisfies the damage threshold. Additionally, in Chapters 4 and 5 we will study distribution topologies minimizing, for all $x \in [n]$, the maximum possible LiSS- resp. LoSS-damage that an attack of cardinality x can impose on them.

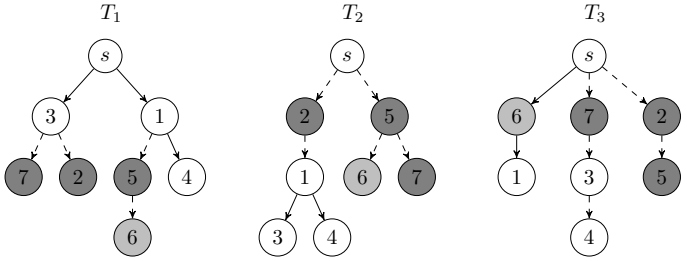
Clearly, it is furthermore possible to study less abstract attack models, which account for more details of practical peer-to-peer streaming systems. Such an approach was followed by a bachelor thesis [Hol10] that was supervised by the author. Here, the properties and applicability of a model explicitly considering bandwidth-exhaustion attacks and the packet loss that is correlated with them were investigated. However, such increasingly complex models are very hard to analyze. Furthermore, their specificity leads to less general results.



(a) LISS-damage



(b) LOSS-damage



(c) FEC-LOSS-damage

Figure 3.1.: Example of damage measures. With $z = 2$, the attack $X = \{2, 5, 7\}$ leads to (a) 16 packets not reaching their destination, (b) nodes $L^{\mathrm{MDC}}_{X,z} = \{2, 3, 4, 5, 6, 7\}$ losing service when using MDC, and (c) nodes $L^{\mathrm{FEC}}_{X,z} = \{2, 5, 6, 7\}$ losing service when using FEC. Attacked nodes are marked dark gray, damaged nodes with a lighter gray.

3.2. Complexity and Approximability of Attack Problems

In this section, we investigate the hardness of computational problems that are posed when planning an attack on a distribution topology. We will see that this hardness is determined by the applied damage function and the parameters of the attacked topology.

A motivation of our approach and the definition of the studied attack problems are given in Subsection 3.2.1. The subsequent Subsection 3.2.2 introduces the necessary background on computational complexity and approximability. Then, the complexity and approximability of the attack problems for LiSS-, LoSS-, and FEC-LoSS-damage is analyzed in the Subsections 3.2.3, 3.2.4, and 3.2.5, respectively.

3.2.1. Attack Problems

The studied attack problems are the essential optimization problems posed to a well-informed but resource-restricted attacker:

> Given a distribution topology and attack parameters including a damage threshold, we want to find an attack of minimum cardinality that satisfies the damage threshold.

Additional to their relevance from a theoretical point of view, the results of such a study give interesting insights for practitioners as well. In particular, we are able to analytically qualify the influence of the damage function, topology parameters, and the applied stream encoding on the difficulty of the problem of planning "good" attacks on a topology. Especially, we show that the respective search versions of the attack problems are **NP**-complete for all damage measures of Section 3.1. Under the assmption $\mathbf{P} \neq \mathbf{NP}$, we identify limitiations on the solution quality that can be guaranteed by attackers restricted to reasonable, i.e., polynomial-time, computing resources. Additionally, we point out polynomial-time algorithms that can at least guarantee certain bounds on the solution quality. All these results are particularly interesting, since they demonstrate that the existence of multiple trees *significantly* hardens the problem of finding "good" attacks on push-based peer-to-peer live streaming systems.

We see that the approach of this section also provides us with insights about desireable topology properties. Analyzing the results, we can identify properties that complicate the planning of a successful attack.

Furthermore, the goal of building attack-stable topologies can sometimes conflict with other topology requirements such as distribution efficiency (see [Str07, BSS09]). Here, by providing a notion of practical difficulty of optimal attacks, it becomes possible to qualify their threat. Thus, we obtain a basis to evaluate necessary trade-offs between topology stability and practicability of a streaming system.

The results of this section are published in [GFBS11].

Let us now formally state the attack optimization problem for each of the damage types from Section 3.1.

Definition 3.2.1 ***Minimum Live Streaming Stability*** (LiSS)
Given a distribution topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ and damage threshold $t \in [kn]$, find an attack $X \subseteq V$ with $a^{\mathcal{T}}(X) \geq t$ and minimum cardinality.

Definition 3.2.2 ***Minimum Local Streaming Stability with MDC*** (LoSS)
Given a distribution topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, a service loss threshold $z \in [k]$, and damage threshold $t \in [n]$, find an attack $X \subseteq V$ with $\mathrm{b}^{\mathcal{T}}(X, z) \geq t$ and minimum cardinality.

Definition 3.2.3 ***Minimum Local Streaming Stability w. FEC*** (FEC-LoSS)
Given a distribution topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, a service loss threshold $z \in [k]$, and damage threshold $t \in [n]$, find an attack $X \subseteq V$ with $\mathrm{bec}^{\mathcal{T}}(X, z) \geq t$ and minimum cardinality.

3.2.2. NPO Problems and (In-)Approximability

Our method of choice to study the above problems will be in terms of **NPO** problems, approximability, and approximation-preserving reductions. In general, we will follow the notation of [Cre97]. For a more thorough introduction into the topic of (in-)approximability, see [ACK+00].

Definition 3.2.4 ***NP Optimization Problem***
An ***NP** Optimization (**NPO**) problem* is a tuple $(I, \mathrm{sol}, \mathrm{m}, \texttt{type})$ such that

- I is a set of *instances* (or *inputs*),
- for $x \in I$, the function $\mathrm{sol}(x)$ specifies the *set of possible solutions*, every solution $y \in \mathrm{sol}(x)$ is polynomially bounded in its length, and, given x and y, the statement $y \in \mathrm{sol}(x)$ can be validity-checked in polynomial-time.
- for $x \in I$ and $y \in \mathrm{sol}(x)$, the function $\mathrm{m}(x, y) \in \mathbb{N}$ specifies the *value of solution* y and is computable in polynomial time.
- $\texttt{type} \in \{\min, \max\}$.

Given an input $x \in I$, the function $\mathrm{opt}(x)$ specifies the *value of an optimal solution* $\mathrm{opt}(x) := \texttt{type}_{y \in \mathrm{sol}(x)} \mathrm{m}(x, y)$.

In the *search version* of an **NPO** problem, the task is to find any solution meeting a threshold on the solution value. This threshold is an *additional* part of the input.

All attack problems defined in Section 3.2.1 are **NPO** problems: Their instances are given by tuples $(\mathcal{T}, t)$ or $(\mathcal{T}, z, t)$, respectively. The set of solutions for an input x containing a damage threshold t is $\mathrm{sol}(x) = \{X \in \mathcal{P}(V) \mid f(X) \geq t\}$ where $f(X)$ corresponds to $a^{\mathcal{T}}(X), \mathrm{b}^{\mathcal{T}}(X, z)$, or $\mathrm{bec}^{\mathcal{T}}(X, z)$. Furthermore, for all problems it holds that $\mathrm{m}(x, X) = |X|$ and $\texttt{type} = \min$. The length of a binary representation of each

possible solution is $O(n \log n)$, the damage functions can be computed by tree traversals in time $O(kn)$ resp. $O(kn^2)$ (for the FEC-LoSS damage measure) and the cardinality function can be evaluated in linear time.

Given $x \in I$, the solutions $\mathrm{sol}(x)$ can be rated by the ratio between their value and the value of an optimal solution.

Definition 3.2.5 ***Approximation Ratio*** R
Given an **NPO** problem $O = (I, \mathrm{sol}, \mathrm{m}, \texttt{type})$, an input $x \in I$ and a solution $y \in \mathrm{sol}(x)$, the *approximation ratio* of y for O on input x is defined as

$$R_O(x, y) := \max\left\{\frac{\mathrm{m}(x, y)}{\mathrm{opt}(x)}, \frac{\mathrm{opt}(x)}{\mathrm{m}(x, y)}\right\}.$$

A deterministic algorithm $\mathcal{A}$ for an **NPO** problem O can be seen as a function mapping inputs to solutions, such that $\forall x \in I\colon \mathcal{A}(x) \in \mathrm{sol}(x)$. A polynomial-time *algorithm* is said to have an *approximation ratio of* r *for* O if

$$\forall x \in I\colon R_O(x, \mathcal{A}(x)) \leq r. \tag{3.8}$$

An **NPO** problem O is called r*-inapproximable* if no polynomial-time algorithm can achieve an approximation ratio of r for O. If this inapproximability is based on the precondition that $\mathbf{P} \neq \mathbf{NP}$ and $r > 1$, these definitions show that the search version of O must be **NP**-hard. Furthermore, it is **NP**-complete, because it is in **NP** since O is in **NPO**.

Using *approximation-preserving reductions* [Cre97], it is possible to relate (in-)approximability results of different **NPO** problems. In our setting, we will always use *strict approximation-preserving reductions*.

Definition 3.2.6 ***Strict Approximation-Preserving Reduction***
Given **NPO** problems $A = (I_A, \mathrm{sol}_A, \mathrm{m}_A, \texttt{type})$ and $B = (I_B, \mathrm{sol}_B, \mathrm{m}_B, \texttt{type})$, a *strict approximation-preserving reduction* $A \leq_{\mathrm{strict}} B$ is a pair (f, g) of polynomial-time computable functions, such that

- $f\colon I_A \to I_B$,
- for all $x \in I_A$ and $y \in \mathrm{sol}_B(f(x))$, it holds that $g(x, y) \in \mathrm{sol}_A(x)$ and

$$R_A(x, g(x, y)) \leq R_B(f(x), y).$$

The basic scheme of such a reduction is illustrated in Figure 3.2.

The following propositions are a direct consequence of the above definitions.

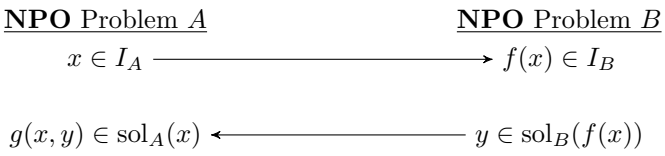


Figure 3.2.: Scheme of an approximation-preserving reduction [Cre97].

Corollary 3.2.7
Let A and B be two **NPO** problems such that $A \leq_{\text{strict}} B$ exists. It holds that:

- if there is an r-approximation algorithm for B, then there is an r-approximation algorithm for A.
- if A is r-inapproximable, then B is r-inapproximable.

We are now ready to analyze the approximability of our attack problems.

3.2.3. (In-)Approximability of the LiSS Problem

The first problem studied is the LiSS problem. We show that its structure is highly similar to the Minimum Partial Set Cover problem.

Definition 3.2.8 ***Minimum Partial Set Cover*** (Min PSC)
Given a ground set U, a set $S \subseteq \mathcal{P}(U)$, and a threshold $t \in [\,|U|\,]$, determine a set $X \subseteq S$ of minimum cardinality such that $\left|\bigcup_{s_i \in X} s_i\right| \geq t$.

W.l.o.g, we can assume that $\bigcup_{s_i \in S} s_i = U$. The Min PSC problem contains the Minimum Set Cover problem as subproblem where $t := |U|$. Under the highly reasonable assumption that $\mathbf{NP} \not\subset \mathbf{DTIME}(O(n^{\log \log n}))$, it is therefore $((1 - o(1)) \ln |U|)$-inapproximable [Fei98]. Under the stronger assumption that $\mathbf{P} \neq \mathbf{NP}$, it is still $c \ln |U|$-inapproximable for a constant $0 < c < 1$ [AMS06] (the paper gives $c = 0.2267$, but a number of preconditions are left unclear).

Theorem 3.2.9
If $\mathbf{P} \neq \mathbf{NP}$, the LiSS problem is $c_1 \log(k)$- and $c_2 \log(n)$-inapproximable for constants $c_1, c_2 > 0$. Here, n is the number of peers and k the number of stripes of the input topology.

Proof. We strictly reduce Min PSC to the LiSS problem.

Given an instance (U, S, t), the reduction function f constructs a distribution topology $\mathcal{T}$ having $k := |U|$ stripes. The node set of $\mathcal{T}$ contains a node s_i for each $s_i \in S$ and dummy nodes $D = \{d_1, \ldots, d_{|U| \cdot |S|}\}$, such that $n = (|U| + 1)|S|$.

For an element $e \in U$, define $S_e := \{s_i \in S \mid e \in s_i\}$ and $S_{\overline{e}} := S \setminus S_e$. The topology $\mathcal{T}$ contains a stripe T_e for each $e \in U$, such that in T_e all nodes $S_{\overline{e}}$ are children of the

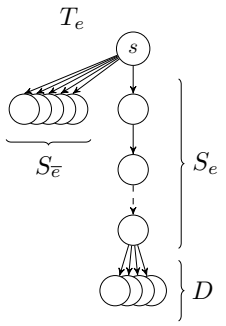


Figure 3.3.: Schematic tree T_e for $e \in U$ built in the reduction Min PSC $\leq_{\text{strict}}$ LiSS.

source and for each $d_i \in D$ we have $\text{pred}_e^{\mathcal{T}}(d_i) = S_e \cup \{s\}$. This definition ensures that the nodes S_e are lined up on each path in T_e that starts at the source and leads to a dummy node. Figure 3.3 gives an example of such a tree. Finally, the function f returns a LiSS instance consisting of topology $\mathcal{T}$ and damage threshold $t \cdot |D|$.

Called on Min PSC instance (U, S, t) and a LiSS-solution $Y \subseteq S \cup D$, the function g returns S if $|Y| \geq |S|$. Otherwise, it iteratively substitutes a dummy $d \in Y \cap D$ in Y by a node $s_i \in S \setminus Y$, while $Y \cap D \neq \emptyset$. Finally, the resulting set Y is returned.

Claim 3.2.10
For every attack $Y \subseteq S \cup D$ on $\mathcal{T}$, it holds that $a^{\mathcal{T}}(g((U, S, t), Y)) \geq a^{\mathcal{T}}(Y)$.

Proof. If $g((U, S, t), Y)$ returns S, all heads $H^{\mathcal{T}}$ are removed and *all* $k(|S| + |D|)$ source-peer paths are lost.

Hence assume $|Y| < |S|$. In this case, g iteratively exchanges some $d \in Y \cap D$ with an $x \in S \setminus Y$. Let Y' be the altered set after such an exchange. We have $\forall e \in U\colon |\text{succ}_e^{\mathcal{T}}(Y')| \geq |\text{succ}_e^{\mathcal{T}}(Y)|$, since for every $T_e \in \mathcal{T}$ one of the following cases holds:

- $S_e \cap Y \neq \emptyset$: It holds that $\text{succ}_e^{\mathcal{T}}(d) \subset \text{succ}_e^{\mathcal{T}}(Y \setminus \{d\}) = \text{succ}_e^{\mathcal{T}}(Y)$. Therefore, we obtain $\text{succ}_e^{\mathcal{T}}(Y') = \text{succ}_e^{\mathcal{T}}(Y) \cup \text{succ}_e^{\mathcal{T}}(x)$.

- $S_e \cap Y = \emptyset$: Since we have $|\text{succ}_e^{\mathcal{T}}(x)| \geq 1, |\text{succ}_e^{\mathcal{T}}(d)| = 1$, and $x \notin \text{succ}^{\mathcal{T}}(Y)$, it follows that $|\text{succ}_e^{\mathcal{T}}(Y')| = |\text{succ}_e^{\mathcal{T}}(Y)| - |\text{succ}_e^{\mathcal{T}}(d)| + |\text{succ}_e^{\mathcal{T}}(x)| \geq |\text{succ}_e^{\mathcal{T}}(Y)|$.

For each exchange, we obtain $a^{\mathcal{T}}(Y') \geq a^{\mathcal{T}}(Y)$ and finally $a^{\mathcal{T}}(g((S, C, t), Y)) \geq a^{\mathcal{T}}(Y)$. □

Claim 3.2.11
For every $Y \subseteq S$, it holds that $\left|\bigcup_{s_i \in Y} s_i\right| \geq t \Leftrightarrow a^{\mathcal{T}}(Y) \geq t \cdot |D|$.

Proof. "⇒": Due to the definition of S_e, it holds that $\forall e \in \bigcup_{s_i \in Y} s_i \colon S_e \cap Y \neq \emptyset$. Hence, the attack Y removes a predecessor of the whole set D in at least t trees of $\mathcal{T}$. We obtain $a^{\mathcal{T}}(Y) \geq t \cdot |D|$.

"⇐": Since $Y \subseteq S$, in each tree $T_e \in \mathcal{T}$ either all or none of the dummy nodes lose a predecessor due to attack Y.

We show that if $a^{\mathcal{T}}(Y) \geq t \cdot |D|$, then the paths to dummy nodes must be lost in at least t trees of $\mathcal{T}$. For this, assume the opposite, i.e., that in at least $k - t + 1$ trees both the set D and the predecessor(s) of D can still be reached from the source. Then we had $a^{\mathcal{T}}(Y) \leq k(|S| + |D|) - (k - t + 1)(|D| + 1) = t|D| - (k - t + 1) < t|D|$, because $k|S| = |U| \cdot |S| = |D|$. This contradicts the assumed damage.

Since $Y \subseteq S$, in a tree T_e the packets of dummies are only lost if $S_e \cap Y \neq \emptyset$. Thus, we obtain $\left|\bigcup_{s_i \in Y} s_i\right| \geq t$. □

Together, the Claims 3.2.10 and 3.2.11 show that $g((U, S, t), Y)$ is a valid solution for Min PSC instance (U, S, t), if Y is a valid solution for our LiSS instance $(\mathcal{T}, t|D|)$. Furthermore, all valid solutions for a Min PSC instance (U, S, t) are valid solutions for LiSS instance $(\mathcal{T}, t|D|)$.

Both problems share a common measure of solution value: the cardinality. Hence, optimum solutions will have an identical value for both problems. For Min PSC instance (U, S, t) and LiSS solution Y on input $f((U, S, t))$, Claim 3.2.10 leads to

$$R_{\text{Min PSC}}((U, S, t), g((U, S, t), Y)) \leq R_{\text{LiSS}}(f((U, S, t)), Y). \tag{3.9}$$

This confirms that (f, g) is indeed a strict approximation-preserving reduction.

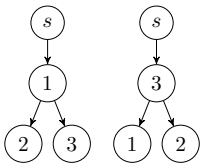
The reduction function f returned topologies with $n = (|U| + 1)|S|$ and $k = |U|$. Due to Corollary 3.2.7, we obtain a $c_1 \log(k)$-inapproximability result for the LiSS problem, for some constant $c_1 > 0$. If $\mathbf{NP} \not\subset \mathbf{DTIME}(n^{O(\log \log n)})$, then c_1 can be set to any constant value smaller than 1.

Note that the Minimum Set Cover problem maintains its logarithmic inapproximability when it is restricted to instances with $|S| \leq |U|^2$. This can be shown by a straight-forward reduction from the Minimum Dominating Set problem. Hence, we also obtain a $c_2 \log(n)$-inapproximability result for some constant $0 < c_2 < \frac{1}{2}$. □

As a consequence of Theorem 3.2.9, not only finding the *optimal* solution for a LiSS instance is **NP**-complete. The search problem even maintains its complexity, if we would be content with solutions that are by a logarithmic factor in the number peers larger than the optimum.

There is also a strict reduction LiSS $\leq_{\text{strict}}$ Min PSC: Given a topology $\mathcal{T}$ of k stripes over node set V and a damage threshold t, the reduction function f returns a Min PSC instance (U, S, t) with ground set

$$U := V \times [k] \tag{3.10}$$



(a) Topology $\mathcal{T}$

$$U = \{(1,1),(1,2),(2,1),(2,2),(3,1),(3,2)\}$$
$$S = \{\{(1,1),(2,1),(3,1),(1,2)\},$$
$$\{(2,1),(2,2)\},$$
$$\{(3,1),(1,2),(2,2),(3,2)\}\}$$

(b) Set system of Min PSC instance (U, S, t).

Figure 3.4.: Set system returned by f in reduction LiSS $\leq_{\text{strict}}$ Min PSC.

and the set of sets

$$S := \{s_v \mid v \in V\} \tag{3.11}$$

with

$$s_v := \bigcup_{i \in [k]} \left(\text{succ}_i^{\mathcal{T}}(v) \times \{i\}\right). \tag{3.12}$$

Figure 3.4 gives an example. Since in each stripe of $\mathcal{T}$ there is a bijection between the peers and their respective successor sets, it holds that $\forall u, v \in V\colon u = v \vee s_u \neq s_v$. Given a Min PSC solution $Y \subseteq S$, the function g returns the set of nodes corresponding to the sets in Y:

$$g((\mathcal{T}, t), Y) := \left\{v \in V \;\middle|\; s_v \in Y\right\}. \tag{3.13}$$

We notice that Y is a solution of Min PSC instance (U, S, t) if and only if $g((\mathcal{T}, t), Y)$ is a solution for LiSS instance $(\mathcal{T}, t)$, i.e., $|\bigcup_{s_v \in Y} s_v| \geq t \Leftrightarrow a^{\mathcal{T}}(g((\mathcal{T}, t), Y)) \geq t$, since

$$\left|\bigcup_{s_v \in Y} s_v\right| = \left|\bigcup_{s_v \in Y} \bigcup_{i \in [k]} \left(\text{succ}_i^{\mathcal{T}}(v) \times \{i\}\right)\right| \tag{3.14}$$

$$= \left|\bigcup_{i \in [k]} \bigcup_{s_v \in Y} \left(\text{succ}_i^{\mathcal{T}}(v) \times \{i\}\right)\right| \tag{3.15}$$

$$= \left|\bigcup_{i \in [k]} \text{succ}_i^{\mathcal{T}}(g((\mathcal{T}, t), Y)) \times \{i\}\right| \tag{3.16}$$

$$= \sum_{i \in [k]} |\text{succ}_i^{\mathcal{T}}(g((\mathcal{T}, t), Y))| \tag{3.17}$$

$$= a^{\mathcal{T}}(g((\mathcal{T}, t), Y)). \tag{3.18}$$

Both problems use cardinality to measure solution value. Since $|Y| = |g((\mathcal{T}, t), Y)|$ holds for each Min PSC solution Y, the values of optimal solutions coincide, too. The functions f and g are polynomial-time-computable. Consequently, the pair (f, g) is a strict approximation-preserving reduction.

Algorithm 1: Greedy algorithm for the LiSS problem

Input: $(\mathcal{T}, t)$
$Y := \emptyset$;
while $a^{\mathcal{T}}(Y) < t$ **do**
  Fix $x \in \arg\max_{x \in V \setminus Y} a^{\mathcal{T}}(Y \cup \{x\}) - a^{\mathcal{T}}(Y)$;
  $Y := Y \cup \{x\}$;
return Y;

Thus, approximation algorithms for the Min PSC problem can be used to approximate LiSS. The most prominent one is the intuitive *greedy algorithm*, that iteratively chooses a set covering the most uncovered elements. This algorithm is well-studied and can guarantee an approximation ratio of $\min(\max_{s_v \in S} |s_v|, H(t))$ [Wol82, Sla97], where $H(x) := \sum_{i=1}^{x} \frac{1}{i}$ is the x-th harmonic number. $H(n)$ is upper bounded by $\ln(x) + 1$.

Due to the above reduction, this greedy algorithm is equivalent to Algorithm 1, the natural greedy algorithm for LiSS.

Corollary 3.2.12
Algorithm 1 is a $\min(\max_{v \in V} a^{\mathcal{T}}(v), H(t))$-approximation algorithm for the LiSS problem.

Over all possible instances, we have $\min(\max_{v \in V} a^{\mathcal{T}}(v), H(t)) \leq H(kn) \leq \ln(kn) + 1$. Furthermore, due to Theorem 3.2.9 and given $\mathbf{P} \neq \mathbf{NP}$, the guaranteed approximation ratio of solutions of Algorithm 1 is by at most a constant factor greater than the guarantees that *any* polynomial-time approximation algorithm for LiSS is able to give.

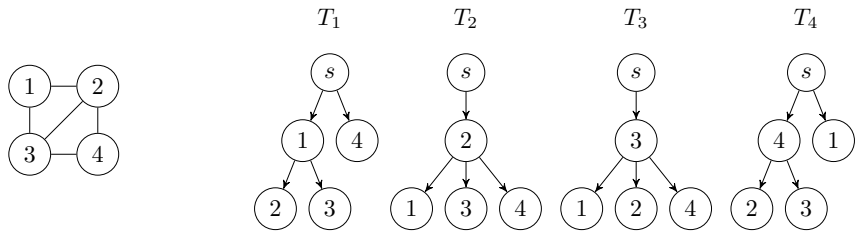
Clearly, it is possible to further improve Algorithm 1. For example, the introduction of a constant lookahead seems promising. However, such enhanced algorithms are equivalently limited by the inapproximability result of Theorem 3.2.9.

Together, both reductions show that the approximability of LiSS is very similar to that of the Minimum Set Cover problem. If $\mathbf{P} \neq \mathbf{NP}$, Minimum Set Cover is complete for the class of logarithmically approximable problems under E-reduction [KMSV99, AMS06]. Since the latter is a generalization of the strict reduction [Cre97], the same applies to the LiSS problem.

It is important to note that all results of this chapter are worst-case results. Consequently, for carefully chosen *sub*sets of LiSS instances, better approximation algorithms will exist. A drastic example is given by the optimally LiSS-stable topologies of Section 4.1, for which a simple polynomial-time greedy algorithm guarantees to find an *optimal* LiSS solution. However, the stability of such topologies is not compromised this fact, since the same algorithm achieves at least an equal value of damage on all other topologies with the same parameters.

3.2.4. (In-)Approximability of the LoSS Problem

Now, we turn to the (in-)approximability of the LoSS problem. We will base our results on the Minimum Dominating Set problem.



(a) Graph G (b) Topology $\mathcal{T}$

Figure 3.5.: Example for topologies built in the reduction Min DS $\leq_{\text{strict}}$ LoSS.

Definition 3.2.13 ***Minimum Dominating Set*** (Min DS)
Given a graph $G = (V, E)$, choose a subset $X \subseteq V$ of minimum cardinality, that satisfies $\forall u \in V \setminus X, \exists v \in X \colon \{u, v\} \in E$.

The approximation properties of the Min DS problem are similar to that of the Minimum Set Cover problem [Kan92]. In particular, it is $c \log |V|$-inapproximable for a constant $c > 0$, if $\mathbf{P} \neq \mathbf{NP}$.

Theorem 3.2.14
If $\mathbf{P} \neq \mathbf{NP}$, the LoSS-problem is $c_1 \log(k)$- and $c_2 \log(n)$-inapproximable for constants $c_1, c_2 > 0$. Here, n is the number of peers and k the number of stripes of the input topology.

Proof. We show a strict reduction Min DS $\leq_{\text{strict}}$ LoSS.

Given a graph $G = (V, E)$, the reduction function f returns a LoSS instance $(\mathcal{T}, z, t)$ with $z := 1$ and $t := |V|$. The topology $\mathcal{T}$ has $k := |V|$ stripes over node set V. For a node $v \in V$, define the *neighborhood of v in graph G* as $N(v) := \{u \in V \mid \{u, v\} \in E\}$. The topology $\mathcal{T}$ contains a stripe T_v for each $v \in V$. In stripe T_v, it holds that $\text{child}_v^{\mathcal{T}}(v) = N(v)$ and $H_v^{\mathcal{T}} = V \setminus N(v)$. Figure 3.5 gives an example of such a construction.

Called on a LoSS solution Y, the function g simply returns Y. Both f and g are polynomial-time computable.

Claim 3.2.15
A set $Y \subseteq V$ is a Min DS solution on G if and only if Y is a LoSS solution for input $(\mathcal{T}, 1, |V|)$.

Proof. Let $Y \subseteq V$ be a LoSS solution for $(\mathcal{T}, 1, |V|)$, i.e., Y contains at least one predecessor in $\mathcal{T}$ for every node in V. Since in $\mathcal{T}$ a node $u \in V$ forwards only in stripe T_u, it holds that $\forall v \in V \; \exists u \in Y \colon v \in Y \vee v \in \text{succ}_u^{\mathcal{T}}(u)$. By construction of $\mathcal{T}$, this is the case if and only if $\forall v \in V \; \exists u \in Y \colon v \in Y \vee v \in N(u)$. Thus, Y is a dominating set of G. □

From Claim 3.2.15 and the fact that MIN DS and LOSS both use cardinality to measure the value of a solution, it follows that (f, g) is indeed a strictly approximation-preserving reduction. Since the built topologies have $k = |V|$ and $n = |V|$, Theorem 3.2.14 follows from the inapproximability of MIN DS. □

In contrast to the LISS problem, we currently do not know whether this approximability bound is tight.

However, as a first step towards a LOSS approximation, we can identify certain sets of LOSS instances for which approximation algorithms are known.

- A trivial case appears as soon as we restrict the number of topology heads to a constant $c \in \mathbb{N}$. In this case, optimal solutions are of cardinality at most c (since $H^{\mathcal{T}}$ is a solution achieving every possible damage threshold), such that the number of solution candidates is polynomial: $\sum_{i=1}^{c} \binom{n}{i} \leq \sum_{i=1}^{c} n^i$. Thus, this subproblem is in **P**.

- If we alternatively restrict to LOSS instances $(\mathcal{T}, z, t)$ with $z = 1$, an attacked node directly damages all its undamaged successors. This subproblem of LOSS can be seen as a MIN PSC instance with $U := V$, $S := \{ \bigcup_{i\in[k]} \mathrm{succ}_i^{\mathcal{T}}(v) \mid v \in V\}$, and an unchanged threshold t. Therefore, it is $\min(\max_{v\in V} \mathrm{b}^{\mathcal{T}}(\{v\}, 1), H(t))$-approximable [Wol82, Sla97].

- If we restrict to topologies of maximum depth 2, the LOSS problem can be interpreted as a PARTIAL MULTI-SET MULTI-COVER problem: the problem of covering t elements of U, each at least z times, and allowing S to be a set of multisets over U. In particular, we had $U = V$ and $S = \{ \biguplus_{i\in[k]} \mathrm{succ}_i^{\mathcal{T}}(v) \mid v \in V \}$. Additionally, a reverse reduction to LOSS seems possible, too. However, approximability results are currently only known for the restriction PARTIAL SET MULTI-COVER, for which a $\frac{4d}{3} + \epsilon$-approximation [RS11] (with $d := \max_{s_i\in S} |s_i|$ and $\epsilon > 0$) was recently found. If $d(\mathcal{T}) = 2$ and $t = n$, we can apply an LP-based approximation algorithm [Kol00, Vaz04] for MULTI-SET MULTI-COVER and achieve an approximation ratio of $H(\max_{v\in V} a^{\mathcal{T}}(v))$ on these instances.

- Restricting to LOSS instances $(\mathcal{T}, z, t)$ with arbitrary topologies and $t = n$, we can apply an algorithm running z rounds of the greedy algorithm for MINIMUM SET COVER.

  Starting with a solution $Y = \emptyset$, in each round the algorithm approximates a set cover instance (U, S) with $U := \{v \in V \mid \mathrm{inc}_Y^{\mathcal{T}}(v) > k-z\}$ and $S = \{s_v \mid v \in V\backslash Y\}$. Here, each s_v contains all nodes $u \in U$ that have an intact $s \to u$ path leading over v after the nodes Y are removed from $\mathcal{T}$.

  After each round, the set cover solution is added to Y and after z rounds, Y is returned. It is a valid LOSS solution since each $v \in V$ has predecessors from at least z stripes in Y. Furthermore, let X be an optimal LOSS solution for $(\mathcal{T}, z, t)$ and let S_i be an *optimal* set cover solution in round i. We have $\sum_{i\in[z]} |S_i| \leq z|X|$,

since in each round $X \setminus Y$ is a valid set cover solution, too. Consequently, this algorithm results in a $zH(n)$-approximation:

$$|Y| \leq H(n) \sum_{i \in [z]} |S_i| \leq zH(n)|X| \tag{3.19}$$

- If the LoSS problem is modified, such that a *specific subset* $Z \subseteq V$ has to be damaged, we can reuse the z-round set cover algorithm described above. In particular, we add the additional restriction that the unimportant nodes $V \setminus Z$ are removed both from U and all sets in S. Due to the arguments given above, this results in a $zH(\,|Z|\,)$-approximation.

Albeit no non-trivial approximation algorithm is currently known for the *general* LoSS problem, it has notable similarities with the family of classical covering problems around Minimum Set Cover. Given $\mathbf{P} \neq \mathbf{NP}$, the latter is the canonical problem for the class of logarithmically approximable **NPO** problems [Hoc97, ACK+00]. Therefore, it seems possible that a logarithmic approximation algorithm for the general LoSS problem exists. A promising source for a confirmation or rejection of such a conjecture are future results on the most similar Set Cover variant, the Partial Multi-Set Multi-Cover problem. However, to date, non-trivial approximability results exist only for the less general Partial Set Multi-Cover [RS11].

Again, it has to be noted that the (in-)approximability results of this section are worst-case results over all possible instances of the LoSS problem. Consequently, the approximability can be better in both the average-case and for specifically chosen instances. A respective indication is given by experimental results [Gum11], comparing the solution value of simple greedy algorithms and exact exponential-time algorithms for the LoSS problem on samples of different classes of distribution topologies. Until the damage threshold was reached, the greedy algorithms iteratively attacked a node being predecessor to the highest number of undamaged nodes in the topology. Although, they performed remarkably bad on most classes, they were close to optimal solutions for topologies consisting of unbalanced trees. In this case, the structure of optimal solutions is clearly favoured by the greedy algorithm's node selection strategy.

3.2.5. Inapproximability of the FEC-LoSS Problem

Finally, we analyze the inapproximability of the FEC-LoSS problem. Here we are able to show considerably higher inapproximability bounds than for the LoSS problem that uses Multiple Description Coding.

The results of this section relate FEC-LoSS with a class of **NPO** problems for which the canonical problem is Label Cover [Hoc97]. This class is believed to be seperate from the logarithmically approximable problems for which Set Cover is complete if $\mathbf{P} \neq \mathbf{NP}$. Label Cover has received attention due to its connections to proof theory [DS04]. Here, however, we use the Red-Blue Set Cover problem, which features equal inapproximability results and a much simpler formulation.

Definition 3.2.16 ***Red-Blue Set Cover*** (RBSC) ***[CDKM00]***
Given two disjoint ground sets R and B of *red* resp. *blue* elements and a set S of sets $s_i \subseteq R \cup B$ with $R \cup B = \bigcup_{s_i \in S} s_i$, find a subset $Y \subseteq S$ with $B \subseteq \bigcup_{s_i \in Y} s_i$ that minimizes $\left|\bigcup_{s_i \in Y} s_i \cap R\right|$.

If $\mathbf{P} \neq \mathbf{NP}$, RBSC cannot be approximated with ratio $2^{\log^{1-\delta}|S|}$, where $\delta = 1/\log\log^c|S|$ for any constant $c < \frac{1}{2}$ [CDKM00]. This result places it beyond the classes of logarithmically (e.g., Set Cover) and polylogarithmically (e.g., Group Steiner Tree) approximable problems. Furthermore, the same inapproximability holds for the variant 2-1-RBSC where S contains only sets of one blue and two red elements [CDKM00].

Theorem 3.2.17
If $\mathbf{P} \neq \mathbf{NP}$, then the FEC-LoSS problem is inapproximable within factors of $2^{\log^{1-o(1)}\Theta(k)}$ and $2^{\log^{1-o(1)}\Theta(\sqrt{n})}$, respectively.

Proof. We show a strict reduction 2-1-RBSC $\leq_{\text{strict}}$ FEC-LoSS.

Called on a 2-1-RBSC instance (R, B, S) the reduction function f constructs a topology $\mathcal{T}$ of $k := 2|S|$ stripes. The node set $V := R \cup \{s_{i,1}, s_{i,2} \mid s_i \in S\} \cup V_B$ contains the *red nodes* R, two *set nodes* $s_{i,1}, s_{i,2}$ per $s_i \in S$, and a set of *blue replica nodes* $V_B := \{v \in V_{b,j} \mid j \in \{1,2\}, b \in B\}$. For each $b \in B$, the latter is organized, into two *replica blocks* $V_{b,j} := \{b_j^q \mid 1 \leq q \leq |R| + 2|S|\}$. Consequently, it holds that $n = (2|B| + 1)(|R| + 2|S|)$.

For each set $s_i = \{r_1, r_2, b\} \in S$, there are two unique stripes T_{r_1,s_i} and T_{r_2,s_i}. In particular, for $q, j \in \{1,2\}$, in T_{r_q,s_i} it holds that $\text{child}^{\mathcal{T}}_{r_q,s_i}(r_q) = \{s_{i,1}, s_{i,2}\}$ and $\text{child}^{\mathcal{T}}_{r_q,s_i}(s_{i,j}) = V_{b,j}$. All other nodes are children of source s. Figure 3.6 gives an example of such a set-specific tree pair. After $\mathcal{T}$ is constructed, f returns FEC-LoSS input $(\mathcal{T}, z, t)$ with $z := 2$ and $t := 2|B|(|R| + 2|S|)$.

Called on 2-1-RBSC instance (R, B, S) and FEC-LoSS solution $Y \subseteq V$, the function g returns S if $|Y| \geq |R|$. Otherwise, g first removes all those set nodes from Y whose blue element can be covered by a set containing only red nodes from Y:

$$Y := Y \setminus \{s_{i,j} \in Y \mid b \in s_i \cap B \wedge r_1, r_2 \in Y \cap R \wedge \{r_1, r_2, b\} \in S\} \tag{3.20}$$

Then, for each $b \in B$, the set $S_{Y,b} := \{s_i \in S \mid b \in s_i \wedge \{s_{i,1}, s_{i,2}\} \cap Y \neq \emptyset\}$ is formed. It holds that $\forall b_1, b_2 \in B\colon b_1 \neq b_2 \Rightarrow S_{Y,b_1} \cap S_{Y,b_2} = \emptyset$, since each set in S contains exactly one blue node. For every non-empty set $S_{Y,b}$, both red nodes of *one* representative set $s_i \in S_{Y,b}$ are added to Y and all set nodes corresponding to sets in $S_{Y,b}$ are dropped from Y. Finally, g returns the set $Z := \{\{r_1, r_2, b\} \in S \mid r_1, r_2 \in Y \cap R\}$.

Both f and g are polynomial-time computable.

Claim 3.2.18
If $Y \subseteq V$ is an FEC-LoSS solution on instance $f((R, B, S))$, then $Z := g((R, B, S), Y)$ is a 2-1-RBSC solution on (R, B, S) with $\left|\bigcup_{s_i \in Z} s_i \cap R\right| \leq |Y|$.

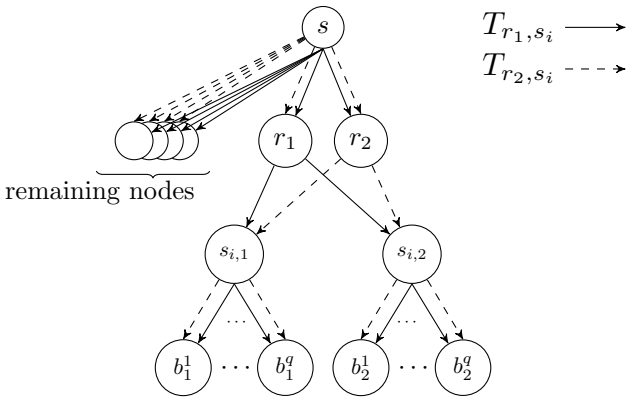


Figure 3.6.: Trees T_{r_1,s_i} and T_{r_2,s_i} for a set $s_i = \{r_1, r_2, b\} \in S$ where $q = |R| + 2|S|$.

Proof. If $|Y| \geq |R|$, the function g returns S, covering both R and B completely. So from now on assume $|Y| < |R|$.

At first note that, since they are heads in each stripe of $\mathcal{T}$, red nodes can only be damaged by being included in the attack Y. Furthermore, all nodes of a blue replica block $V_{b,j}$ have the same predecessors. Hence, without being directly attacked, blue replica nodes can only be damaged in whole blocks $V_{b,j}$ of $|R| + 2|S|$ nodes.

We show that attacking $Y \setminus V_B$ must damage *all* blue replica nodes: Assume there is a block $V_{b,j}$ left undamaged by an attack on $Y \setminus V_B$. It holds that

$$t = 2|B|(|R| + 2|S|) \leq \mathrm{bec}^{\mathcal{T}}(Y, 2) \tag{3.21}$$
$$\leq |V| - |V_{b,j}| + |V_{b,j} \cap Y| - |R \setminus Y| \tag{3.22}$$
$$= 2|B|(|R| + 2|S|) + |V_{b,j} \cap Y| - |R \setminus Y|, \tag{3.23}$$

leading to

$$|R \setminus Y| \leq |V_{b,j} \cap Y| \tag{3.24}$$
$$\Rightarrow |R| = |R \cap Y| + |R \setminus Y| \leq |R \cap Y| + |V_{b,j} \cap Y| \leq |Y|. \tag{3.25}$$

Inequality (3.25) holds because $R \cap V_{b,j} = \emptyset$. It contradicts the assumption $|Y| < |R|$.

For an arbitrary $b \in B$, we study which conditions on Y ensure that *all* nodes $V_{b,1} \cup V_{b,2}$ are damaged by $Y \setminus V_B$. This happens if they lose at least $z = 2$ stripes. The following cases are possible:

- $\exists r_1, r_2 \in Y \colon s_i = \{r_1, r_2, b\} \in S$: In both T_{r_1,s_i} and T_{r_2,s_i} the parents of the set nodes $s_{i,1}$ and $s_{i,2}$ are disabled. Hence, both set nodes are damaged. In particular, they cannot supply their children $V_{b,1} \cup V_{b,2}$ in these stripes. Those are damaged, too.

- $\exists s_{i,1}, s_{j,2} \in Y\colon b \in s_i \wedge b \in s_j$: Let r_1, r_2 be the red nodes in set s_i. Disabling $s_{i,1}$, the nodes $V_{b,1}$ lose their parents in the stripe T_{r_1,s_i} and T_{r_2,s_i}. Hence, they are damaged. The argument for $s_{j,2}$ and $V_{b,2}$ is analogue.

We denote these cases as the *red-nodes case* and the *set-nodes case*, respectively. We also show, that there are no further ways to damage all nodes $V_{b,1} \cup V_{b,2}$ by attacking $Y \setminus V_B$. Assuming that none of the above conditions applies, for each set $s_i = \{r_1, r_2, b_s\} \in S$, distinguish the following cases:

- $b_s \neq b$: The nodes $V_{b,1} \cup V_{b,2}$ are heads in the stripes T_{r_1,s_i} and T_{r_2,s_i}. They cannot lose these stripes due to attack $Y \setminus V_B$.

- $b_s = b \wedge |\{r_1, r_2\} \cap Y| \leq 1$: The nodes $s_{i,1}$ and $s_{i,2}$ are heads in all stripes but T_{r_1,s_i} and T_{r_2,s_i}. Assuming that none of them is attacked, both have at least $k-1$ intact parents and can forward to $V_{b,1} \cup V_{b,2}$ in both stripe T_{r_1,s_i} and T_{r_2,s_i}. However, if $s_{i,1}$ or $s_{i,2}$ are attacked, either the nodes $V_{b,1}$ or $V_{b,2}$ are damaged.

Since the set-nodes case was excluded, damage due to attacked set nodes can happen only for either $V_{b,1}$ or $V_{b,2}$. Hence, at least the other blue replica block is supplied in all stripes. This contradicts that all nodes of $V_{b,1} \cup V_{b,2}$ were damaged.

Consequently, we now know that attack $Y \setminus V_B$ damages all nodes V_B and that, for each blue replica block, this is achieved by either the red-nodes case or the set-nodes case.

Now, trace the computation of g. The initial removal of set nodes cannot increase the cardinality of Y. Note that, for each removed set node $s_{i,j}$ with blue element $b \in s_i$, $Y \setminus V_B$ damages the blue replica nodes $V_{b,1} \cup V_{b,2}$ due to the red-nodes case. In the second step, the sets $S_{Y,b}$ are formed. Since all other set nodes were already removed, these sets are non-empty only for blue elements whose blue replica nodes are damaged only by the set-nodes case. For each b with $|S_{Y,b}| \geq 1$, g adds both red nodes of one set in $S_{Y,b}$ to Y and removes at least the 2 set nodes from the set-nodes condition. Thus, Y cannot grow and after this transformation each blue replica node is damaged due to red nodes in Y. Hence, $Y \cap R$ is a valid solution for our LoSS instance.

Finally, based on the modified Y, g returns $Z = \{\{r_1, r_2, b\} \in S \mid r_1, r_2 \in Y \cap R\}$. It holds that $\left|\bigcup_{s_i \in Z} s_i \cap R\right| \leq |Y \cap R| \leq |Y|$. Since $Y \cap R$ is a valid LoSS-solution damaging $V_{b,1} \cup V_{b,2}$ for all $b \in B$, we have $\forall b \in B\ \exists r_1, r_2 \in Y \cap R\colon \{r_1, r_2, b\} \in S$. Hence, the set Z is a red-blue set cover for (R, B, S). □

Claim 3.2.18 shows that g returns valid 2-1-RBSC solutions for input (R, B, S) and cannot increase solution cardinality.

A similar transformation produces FEC-LoSS solutions from 2-1-RBSC solutions: Given a 2-1-RBSC solution X for instance (R, B, S), an attack $Y = \bigcup_{s_i \in X} s_i \cap R$ on $\mathcal{T}$ disconnects all nodes V_B. In particular, for all $b \in B$, the nodes $V_{b,1} \cup V_{b,2}$ are damaged due to the red-nodes case. Therefore, it holds that $\mathrm{bec}^{\mathcal{T}}(Y, 2) \geq 2|B|(|R| + 2|S|)$ and Y is a valid FEC-LoSS solution for input $f((R, B, S))$. Furthermore, X and Y have equal solution quality.

It follows that optimal solutions for every 2-1-RBSC instance (R, B, S) and the corresponding FEC-LoSS instance $f((R, B, S))$ have a common value opt. For every FEC-LoSS solution Y on input $f((R, B, S))$, we obtain

$$R_{\text{2-1-RBSC}}((R, B, S), g((R, B, S), Y)) = \frac{\left|\bigcup_{s_i \in g((R,B,S),Y)} s_i \cap R\right|}{\text{opt}} \tag{3.26}$$

$$\leq \frac{|Y|}{\text{opt}} = R_{\text{FEC-LoSS}}(f(R, B, S), Y). \tag{3.27}$$

Consequently, (f, g) is a strict approximation-preserving reduction. The built topology $\mathcal{T}$ has the properties $k = 2|S|$ and $n = (2|B| + 1)(|R| + 2|S|) = \Theta(|S|^2)$. Combining these with the inapproximability of 2-1-RBSC, we obtain the stated inapproximability bounds for FEC-LoSS. □

Similar to the LoSS problem, it is an open question whether this bound is tight. In particular, no non-trivial approximation algorithm for FEC-LoSS is currently known.

3.3. Summary

In this chapter, we began studying the influence of distribution topologies on the attack-stability of push-based peer-to-peer live streaming systems. In particular, Section 3.1 introduced and motivated our model of attacks and their consequences for the streaming system. For this, the measures of LiSS-, LoSS-, and FEC-LoSS-damage were defined. In presence of an attack, they count the system-wide number of lost packets and the number of nodes that lost at least a given fraction of stripes when MDC or FEC stream encoding is applied. In Section 3.2, we used these damage measures to formalize the corresponding attacker problems LiSS, LoSS, and FEC-LoSS. In each, the task is to create an aspired amount of damage by attacking a minimum number of peers. We showed that the search versions of all three problems are **NP**-complete and studied their approximability.

Under the assumption that $\mathbf{P} \neq \mathbf{NP}$, the results showed $c_1 \log(k)$- and $c_2 \log(n)$-inapproximability (with constants $c_1, c_2 > 0$) for both the LiSS and the LoSS problem. Furthermore, we proved the $2^{\log^{1-o(1)} \Theta(k)}$ resp. $2^{\log^{1-o(1)} \Theta(\sqrt{n})}$−inapproximability of the FEC-LoSS problem. Hence, there are topologies on which a deterministic attacker with only polynomial computational resources has to attack a number of peers that is at least by these factors larger than the actual minimum required. To the knowledge of the author, these are the first lower bounds on the guaranteed quality of attacks of polynomial-time attackers on peer-to-peer distribution topologies. The results were published in [GFBS11].

Besides these inapproximability results, we also observed that the natural greedy algorithm for the LiSS problem gives a $\min(\max_{v \in V} a^{\mathcal{T}}(v), H(t))$-approximation. Thus, the approximation ratio of its solutions is by at most a constant factor greater than the guarantees that *any* polynomial-time approximation algorithm for LiSS is able to give, if $\mathbf{P} \neq \mathbf{NP}$.

Albeit we could identify a number of special cases in which the LoSS problem is logarithmically approximable, its general approximability remained unclear. Due to its similarities with the Partial Multi-Set Multi-Cover problem, it is to expect that future results on this very general variant of Set Cover can be transfered to the LoSS problem.

A similar open question was posed by the approximability of the FEC-LoSS problem.

Studying the presented reductions, we generally saw that the number of topology stripes directly influences the inapproximability of our attacker problems. Although we have to be careful not to over-simplify the consequences of this observation, it can be said that a high number of stripes offers the *possibility* to construct topologies for which resource-efficient attacks are hard to find.

Finally, it is important to emphasize that all results of Section 3.2 are based on worst-case analysis. Hence, there will be subclasses of topologies which are much easier to attack. Here, the optimally LiSS-stable topologies of Chapter 4 are a prominent example, where the LiSS problem can be solved in **P**. Studying the hardness of attacks on other restricted classes of topologies offers interesting topics for future research.

The two following chapters will now take an opposite approach on attack-stability. Instead of asking for the hardness of finding efficient attacks, we aim at constructing topologies that minimize the maximum damage that attacks of any given set of attack parameters (e.g., cardinality, service loss threshold) can achieve on them. In particular, Chapter 4 will consider the LiSS-damage measure, while Chapter 5 will study the LoSS-damage measure.

4. LiSS-Stability and Topology Construction Rules

After the study of attack problems in Section 3.2, we now want to approach the concept of attack-stability from a very defensive point-of-view. This is motivated by the fact that, although they may be difficult to plan, we have to acknowledge that attacks on peer-to-peer streaming systems can generally not be averted.

However, the peer-to-peer streaming system can be prepared. Especially, the available control over the own distribution topology can be used to form attack-stable topologies. In this process, the following questions are posed:

- Given the current system parameters (e.g., stripe number, source capacity, peer number), are there topologies that minimize the maximum possible damage that is achievable on them for every set of attack parameters?
- If these topologies exist, what are their properties and how are they built?

In the current and the following chapter, we will study these questions when measuring the consequences of attacks in LiSS- and LoSS-damage, respectively. This determination will turn out to heavily influence the requirements for such stable topologies as well as the practicability of constructing them.

We now focus on topologies minimizing possible LiSS-damage. Section 4.1 characterizes optimally LiSS-stable topologies, presents a first subclass whose topologies are easy to construct, and analyzes properties necessary due to their characterization. Based on these findings, a more general subclass is presented in Section 4.2. It is defined by a small set of rules. The following Section 4.3 focusses on studying an important special case, the optimally LiSS-stable head topologies. Section 4.4 shows that recognizing a given topology as optimally LiSS-stable is a **coNP**-complete problem and Section 4.5 sketches possible heuristics to construct the identified classes using a distributed topology management. The chapter is summarized in Section 4.6.

4.1. Optimally LiSS-Stable Topologies

In the following, we give a first introduction to optimally LiSS-stable topologies. In particular, Subsection 4.1.1 presents and motivates their definition. Furthermore, the problems of finding and recognizing optimally LiSS-stable topologies are formalized.

Next, Subsection 4.1.2 studies a greedy attack strategy on distribution topologies. The damage achieved by such attacks is *lower* bounded by sums over a specific damage sequence. By introducing a topology class where maximum possible LiSS-damage has

exactly this *upper* bound, Subsection 4.1.3 gives both a damage-based characterization of optimally LiSS-stable topologies and a first subclass. In particular, testing membership for this subclass is possible in polynomial-time. Based on the obtained characterization, Subsection 4.1.4 then deduces properties necessary for optimally LiSS-stable topologies.

4.1.1. The Problem of Finding Optimally LiSS-Stable Topologies

In our study, we again follow a worst-case approach and minimize only *maximum* possible damage. The alternative would be to optimize topologies such that they minimize the consequences of *individual attack strategies*. However, here a multitude of strategies is thinkable and a topology that is optimal towards one strategy can be quite vulnerable towards another one. We approach possible objections in two ways. On the one hand, by minimizing the damage dealt by an optimal attacker (with exponential-time computing resources), we clearly upper-limit the damage of *all* attack strategies. On the other hand, Section 4.1.2 shows that the LiSS-damage of an optimal attacker on an optimally LiSS-stable topology in $\mathbb{T}(n, C, k)$ can be caused on *every* topology from $\mathbb{T}(n, C, k)$ by a simple greedy attacker. Hence, we can limit the maximum damage on the topology to a level which cannot be prevented and is furthermore easily obtained on every topology. These arguments formed an integral part in the publication [BBG+09].

Based on the goals given above, we now formalize our demands on optimally LiSS-stable topologies.

Definition 4.1.1 ***Optimally* LiSS-*stable Topology***
Given $n, C, k \in \mathbb{N}$ with $n \geq Ck$, a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is called *at least as* LiSS-*stable as* $\mathcal{C} \in \mathbb{T}(n, C, k)$, if it holds that

$$\forall x \in [n] \colon \max_{X \subseteq V, |X| = x} a^{\mathcal{T}}(X) \leq \max_{X \subseteq V, |X| = x} a^{\mathcal{C}}(X).$$

$\mathcal{T}$ is called an *optimally* LiSS-*stable topology*, if it is at least as LiSS-stable as every $\mathcal{C} \in \mathbb{T}(n, C, k)$.

With this definition, we can formalize our problem of obtaining optimally LiSS-stable topologies.

Definition 4.1.2 ***Optimally* LiSS-*stable Topology Formation Problem***
Given $n, C, k \in \mathbb{N}$ with $n \geq Ck$, find an optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ or determine that none exists.

Since the size of the binary representation of topology $\mathcal{T}$ (e.g., an $n \times k$ matrix of predecessors) must be polynomial in the parameters n, C, k and since we will see in Section 4.1.3 that optimally LiSS-stable topologies actually exist in each class $\mathbb{T}(n, C, k)$, computing a solution for this problem will have at least pseudopolynomial runtime.

If we are not only interested in the existence of *any* optimally LiSS-stable topology in $\mathbb{T}(n, C, k)$ but want to decide whether a *given* topology has this property, we obtain the following problem.

Definition 4.1.3 LiSS-*Stability Decision Problem*
Given a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, decide whether $\mathcal{T}$ is optimally LiSS-stable in $\mathbb{T}(n, C, k)$ or not.

Although the Sections 4.1.2-4.3 will identify a number of very important and applicable cases where the LiSS-Stability Decision Problem can be solved in polynomial time, we will prove in Section 4.4 that it is actually **coNP**-complete. Thus, if $\mathbf{P} \neq \mathbf{NP}$, we cannot expect it to be generally solved by polynomial-time algorithms.

4.1.2. A Successful Attack Strategy

In Section 3.2 we have seen that, using the LiSS-damage measure, the planning of resource-efficient attacks on distribution topologies is **NP**-complete in its search version. The same applies to the corresponding maximization problem where the input consists of a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ and attack size $x \in [n]$. The task is to find an attack X on $\mathcal{T}$ with $|X| = x$ that maximizes $a^{\mathcal{T}}(X)$. Here, **NP**-hardness follows from the fact that if a polynomial-time algorithm $\mathcal{A}$ for this problem would exist, we could solve the LiSS problem in polynomial time using binary search and $\log n$ calls to $\mathcal{A}$.

Nonetheless, it is possible to identify a simple polynomial-time greedy algorithm achieving a LiSS-damage lower bounded by sums over a specific damage sequence. This sequence depends on the parameters of $\mathcal{T}$ and the attack size x. As we will see in Section 4.1.3, the lower bound is sharp: it coincides with the maximum damage of an x-node attack on optimally LiSS-stable topologies.

The greedy algorithm first appeared in [BSS09] and is given by Algorithm 2.

Algorithm 2: A successful attack strategy concering LiSS-damage

Input: Topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, attack size limit $x \in [0, n]$

if $x \geq |H^{\mathcal{T}}|$ **then return** $H^{\mathcal{T}}$;
$(H^{\mathcal{T}}_{i_1}, \ldots, H^{\mathcal{T}}_{i_k}) :=$ sort $(H^{\mathcal{T}}_1, \ldots, H^{\mathcal{T}}_k)$ in order of non-decreasing cardinality;
$X := \emptyset$; $j := 1$;
while $|X \cup H^{\mathcal{T}}_{i_j}| \leq x$ **do** $X := X \cup H^{\mathcal{T}}_{i_j}$; $j := j + 1$;
while $|X| < x$ **do** Fix $h \in \arg\max_{v \in H^{\mathcal{T}}_{i_j} \setminus X} a^{\mathcal{T}}(X \cup \{v\})$; $X := X \cup \{h\}$;
return X;

For each but at most one stripe of topology $\mathcal{T}$, the returned attack contains either all heads or none. In particular, the heads of stripes with few heads are attacked with preference. If there is a stripe for which only a subset of heads is attacked, these are chosen greedily one after another. Each chosen head maximizes the increase of LiSS-damage of the resulting attack set. Attacks returned by Algorithm 2 contain $\min(x, |H^{\mathcal{T}}|)$ nodes.

To lower-bound the damage of such an attack, we introduce the following sequence.

Definition 4.1.4 ***Damage Sequence*** $\delta_i^{C,k}$
For fixed values of $n, C, k \in \mathbb{N}$ and with $i \in [Ck]$, $l := \lfloor (i-1)/C \rfloor$, and $h := (i-1 \mod C)$ define the *damage sequence*

$$\delta_i^{C,k} := \begin{cases} \lceil \frac{n}{C} \rceil + (k - 2l - 1) & \text{, if } h < n \mod C \\ \lfloor \frac{n}{C} \rfloor + (k - 2l - 1) & \text{, if } h \geq n \mod C. \end{cases}$$

For growing values of i, the damage sequence $(\delta_i^{C,k})_{1 \leq i \leq Ck}$ is monotonic decreasing. Figure 4.1 shows the plot of an example sequence.

For $q \in [k]$, it holds that

$$\sum_{i=1}^{Cq} \delta_i^{C,k} = q\left(C\left\lfloor \frac{n}{C} \right\rfloor + (n \mod C)\right) + Cq(k-1) - 2\sum_{i=1}^{Cq} \left\lfloor \frac{i-1}{C} \right\rfloor \tag{4.1}$$

$$= qn + Cq(k-1) - 2C\sum_{i=1}^{q-1} i \tag{4.2}$$

$$= qn + Cq(k-1) - 2C\frac{q(q-1)}{2} \tag{4.3}$$

$$= qn + Cq(k-q). \tag{4.4}$$

Furthermore, for $n \geq Ck$ and $i \in [Ck]$, we have

$$\begin{aligned} \delta_i^{C,k} &\geq \left\lfloor \frac{n}{C} \right\rfloor + (k - 2l - 1) = (k-l) + \left\lfloor \frac{n}{C} \right\rfloor - l - 1 \\ &\geq (k-l) + (k - (k-1) - 1) = (k-l). \end{aligned} \tag{4.5}$$

Lemma 4.1.5 ***[BSS09]***
For each $\mathcal{T} \in \mathbb{T}(n, C, k)$ and $x \in [0, n]$, an attack X returned by Algorithm 2 has the property that

$$a^{\mathcal{T}}(X) \geq \sum_{i=1}^{\min(x, Ck)} \delta_i^{C,k}. \tag{4.6}$$

Proof. The proof follows the ideas presented in [BSS09].

If $x \geq |H^{\mathcal{T}}|$, Algorithm 2 returns $X = H^{\mathcal{T}}$. Attacking X disturbs *all* source-peer paths, i.e., it holds that $a^{\mathcal{T}}(X) = kn$. We obtain $a^{\mathcal{T}}(X) = \sum_{i=1}^{Ck} \delta_i^{C,k} \geq \sum_{i=1}^{\min(x,Ck)} \delta_i^{C,k}$, due to Equation (4.4).

From now on assume $x < |H^{\mathcal{T}}|$. Then there is $q \in [0, k-1]$, such that $|\bigcup_{j=1}^{q} H_{i_j}^{\mathcal{T}}| \leq x$ and $|\bigcup_{j=1}^{q+1} H_{i_j}^{\mathcal{T}}| > x$. We define $H_X := \bigcup_{j=1}^{q} H_{i_j}^{\mathcal{T}}$ and $H_X^+ = H_X \cup H_{i_{q+1}}^{\mathcal{T}}$. Algorithm 2 returns an $X \subseteq H^{\mathcal{T}}$ with $H_X \subseteq X$ and $|X| = x$. Since X contains all heads of q stripes and since each of these also loses its path from the source in the remaining stripes, we obtain $a^{\mathcal{T}}(X) \geq qn + (k-q)|X|$.

Assuming that $x \in \{|H_X|, Cq\}$, it holds that

$$a^{\mathcal{T}}(X) \geq qn + (k-q)x \tag{4.7}$$

$$\geq \sum_{i=1}^{Cq} \delta_i^{C,k} - (Cq - x)(k-q) \tag{4.8}$$

$$\geq \sum_{i=1}^{Cq} \delta_i^{C,k} - \sum_{i=x+1}^{Cq} \delta_i^{C,k} = \sum_{i=1}^{x} \delta_i^{C,k}. \tag{4.9}$$

The step from Term (4.7) to (4.8) uses Equation (4.4). The step from Term (4.8) to (4.9) is based on Inequality (4.5), showing that $\delta_1^{C,k}, \ldots, \delta_{Cq}^{C,k} > (k-q)$. Furthermore, we use that $x = |X| \leq \sum_{j=1}^{q} |H_{i_j}^{\mathcal{T}}| \leq Cq$ is true. For $x = |H_X|$, it holds due to Lemma A.0.3, since the stripe head sets are added to X in order of non-decreasing cardinality and since, for every $\mathcal{T} \in \mathbb{T}(n, C, k)$, we have $\sum_{i \in [k]} |H_i^{\mathcal{T}}| \leq Ck$.

We see that Inequality (4.6) holds for all values of $q \in [0, k]$, i.e., for choices of $x \in \{0, C, 2C, \ldots, Ck\}$ or $x \in \{|\bigcup_{j=1}^{1} H_{i_j}^{\mathcal{T}}|, \ldots, |\bigcup_{j=1}^{k} H_{i_j}^{\mathcal{T}}|\}$. Furthermore, for each $0 \leq x < |H^{\mathcal{T}}|$, there are numbers a, b with

$$a = \max(|H_X|, \left\lfloor \frac{x}{C} \right\rfloor C) \leq x < \min(|H_X^+|, \left\lceil \frac{x}{C} \right\rceil C) = b. \tag{4.10}$$

Note that the damage subsequence $(\delta_{a+1}^{C,k}, \ldots, \delta_b^{C,k})$ is non-increasing and satisfies

$$\forall i \in [a+1, b] \colon \delta_i^{C,k} \in \left\{ \left\lfloor \frac{n}{C} \right\rfloor + k + 2 \left\lfloor \frac{x-1}{C} \right\rfloor - 1, \left\lceil \frac{n}{C} \right\rceil + k + 2 \left\lfloor \frac{x-1}{C} \right\rfloor - 1 \right\}.$$

Let $h_{a+1}, \ldots, h_x$ be the last $(x-a)$ nodes in the greedy ordering of $H_X^+ \setminus H_X$ that Algorithm 2 adds to X in its next-to-last line. Note that $X_a := X \setminus \{h_{a+1}, \ldots, h_x\}$ would be the output of Algorithm 2 when run with parameter $x = a$. Furthermore, let $h_{x+1}, \ldots, h_b$ be the $(b-x)$ next nodes in the continuation of this greedy ordering over the remaining, unchosen nodes of $H_X^+ \setminus X$. Defining $X_i := X_a \cup \{h_{a+1}, \ldots, h_i\}$ for $a+1 \leq i \leq b$, we also see that X_i would be the output of Algorithm 2 when run with parameter $x = i$. Additionally, due to our observations above, we know that both $a^{\mathcal{T}}(X_a) \geq \sum_{i=1}^{|X_a|} \delta_i^{C,k}$ and $a^{\mathcal{T}}(X_b) \geq \sum_{i=1}^{|X_b|} \delta_i^{C,k}$ hold.

For $i \in [a+1, b]$, define $\Delta_i := a^{\mathcal{T}}(X_i) - a^{\mathcal{T}}(X_{i-1})$. Clearly, for $i \in [a, b]$, we have $a^{\mathcal{T}}(X_i) = a^{\mathcal{T}}(X_a) + \sum_{j=a+1}^{i} \Delta_j$ and especially $a^{\mathcal{T}}(X_a) + \sum_{j=a+1}^{b} \Delta_i = a^{\mathcal{T}}(X_b)$. Due to the greedy ordering, the sequence $(\Delta_{a+1}, \ldots, \Delta_b)$ is non-increasing. Applying Lemma A.0.1 for the sequences $(\Delta_{a+1}, \ldots, \Delta_b)$ and $(\delta_{a+1}^{C,k}, \ldots, \delta_b^{C,k})$, we obtain

$$a^{\mathcal{T}}(X) = a^{\mathcal{T}}(X_x) = a^{\mathcal{T}}(X_a) + \sum_{j=a+1}^{x} \Delta_j \geq \sum_{j=1}^{a} \delta_j^{C,k} + \sum_{j=a+1}^{x} \delta_j^{C,k} = \sum_{j=1}^{x} \delta_j^{C,k}. \tag{4.11}$$

□

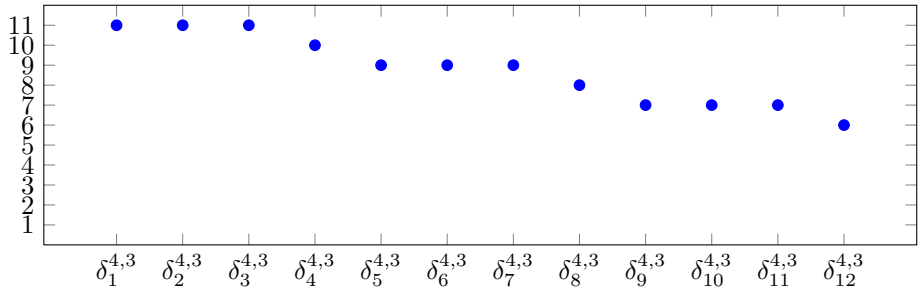


Figure 4.1.: Plot of damage sequence $(\delta_i^{C,k})_{1\leq i\leq 12}$ for $C = 4$, $k = 3$ and $n = 35$.

As we will see in the following subsection, the damage guarantees that Lemma 4.1.5 gives for Algorithm 2 are exactly the values of damage that cannot be averted for attacks of x nodes on topologies from $\mathbb{T}(n, C, k)$. Especially, they coincide with the maximum possible damage that is achievable on the optimally LiSS-stable topologies in $\mathbb{T}(n, C, k)$.

4.1.3. Characterization of Optimally LiSS-Stable Topologies

Next, we study a certain subclass of the topologies $\mathbb{T}(n, C, k)$. It first appeared in [BSS09] (in a disguised form).

Definition 4.1.6 ***Cluster Topologies***
A topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is called *Cluster Topology*, if it can be constructed by the following steps:

1. Partition V into sets $V_1, \ldots, V_C$ with $\forall i \in [C]\colon |V_i| \in \{\lceil n/C \rceil, \lfloor n/C \rfloor\}$.
2. Form topology $\mathcal{T}$ such that for each stripe tree $T_i \in \mathcal{T}$, it holds that
   a) $\forall j \in [C]\ \forall v \in V_j\colon \mathrm{pred}_i^{\mathcal{T}}(v) \subseteq \{s\} \cup V_j$ and
   b) $\forall v \in V\ \forall j \in [k] \setminus \{i\}\colon |\mathrm{succ}_i^{\mathcal{T}\rightarrow}(v)| > 0 \Rightarrow |\mathrm{succ}_j^{\mathcal{T}\rightarrow}(v)| = 0$.

The node set V of a Cluster Topology is split into C subsets (or *clusters*) of nearly equal cardinality. All connections between nodes of different subsets are prohibited. Additionally, for every node $v \in V$, there is at most one stripe in which v is head or has children. Figure 4.2 gives an impression of a Cluster Topology.

Cluster Topologies have a number of interesting properties. Since the clusters V_j are pairwise disjoint and since connections between different clusters are forbidden, the sets $H^{\mathcal{T}} \cap V_j$ for $j \in [C]$ are a partition of $H^{\mathcal{T}}$. The nodes of each V_j are reachable from the source in all k trees, enforcing that each cluster contains a head from each stripe:

$$\forall i \in [k]\ \forall j \in [C]\colon |V_j \cap H_i^{\mathcal{T}}| \geq 1. \tag{4.12}$$

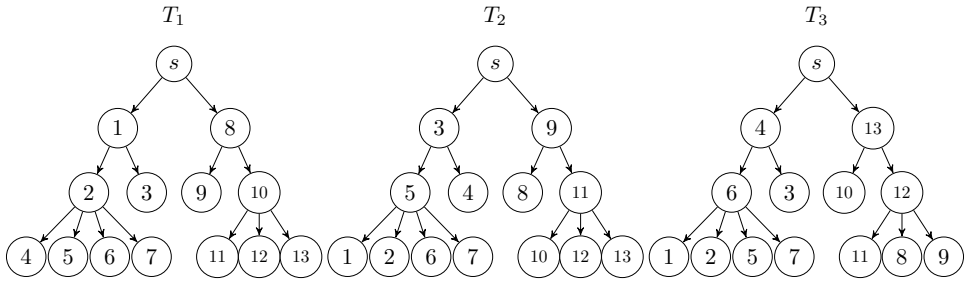


Figure 4.2.: A Cluster Topology from $\mathbb{T}(13, 2, 3)$ with clusters $V_1 = [7]$, $V_2 = [8, 13]$.

Topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ has $|H^{\mathcal{T}}| \leq Ck$. Since no node can be head in more than one stripe, we obtain

$$|H^{\mathcal{T}}| = Ck \text{ and } \forall i \in [k] \colon |H_i^{\mathcal{T}}| = C. \tag{4.13}$$

For each cluster V_j, the induced stripe subtrees $T_i[\{s\} \cup V_j]$ with $i \in [k]$ form a set of k rooted trees with pairwise disjoint sets of forwarding nodes.

Note that all classes $\mathbb{T}(n, C, k)$ have $n \geq Ck$. Hence, we can always form cluster partitions with $\forall j \in [C] \colon |V_j| \geq k$. It is then possible to construct topologies having a unique forwarding peer in each induced stripe subtree $T_i[\{s\} \cup V_j]$. Consequently, Cluster Topologies exist in every class $\mathbb{T}(n, C, k)$.

Now, we study attacks on Cluster Topologies.

Lemma 4.1.7
For every attack $X \subseteq V$ on a Cluster Topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, there is an attack $Y \subseteq H^{\mathcal{T}}$ with $|Y| \leq |X|$ and $a^{\mathcal{T}}(Y) \geq a^{\mathcal{T}}(X)$.

Proof. For each $j \in [C]$, choose $Y_j \subseteq H^{\mathcal{T}} \cap V_j$ with $|Y_j| = \min(k, |X \cap V_j|)$ such that we have $\forall v \in X \cap V_j \; \forall i \in [k] \colon |\mathrm{succ}_i^{\mathcal{T}\rightarrow}(v)| > 0 \Rightarrow H_i^{\mathcal{T}} \cap V_j \subseteq Y_j$. Hence, Y_j contains *at least* the heads supplying the set V_j in stripes in which a node of $X \cap V_j$ is forwarding in. Since $|H^{\mathcal{T}} \cap V_j| = k$ and since each node forwards in at most one stripe, the cardinality limit for Y_j can always be satisfied. Then form Y as $Y := \bigcup_{j \in [C]} Y_j$. The sets $X \cap V_j$ for all $j \in [C]$ are a partition of X. Therefore, it holds that $|Y| \leq \sum_{j \in [k]} |X \cap V_j| = |X|$.

Since there are no successor relationships between nodes of different clusters, we can write for every attack $Z \subseteq V$:

$$a^{\mathcal{T}}(Z) = \sum_{i \in [k]} |\mathrm{succ}_i^{\mathcal{T}}(Z)| = \sum_{j \in [C]} \sum_{i \in [k]} |\mathrm{succ}_i^{\mathcal{T}}(Z \cap V_j)| \tag{4.14}$$

Attacking Y, one of the following cases applies for all $i \in [k]$ and all $j \in [C]$:

1. $H_i^{\mathcal{T}} \cap V_j \subseteq Y$: Attack Y contains the (single) predecessor of all nodes V_j in T_i. This leads to $|\mathrm{succ}_i^{\mathcal{T}}(Y \cap V_j)| = |V_j|$, which is the maximum possible value of LiSS-damage to the nodes V_j in stripe i. Hence, $|\mathrm{succ}_i^{\mathcal{T}}(Y \cap V_j)| \geq |\mathrm{succ}_i^{\mathcal{T}}(X \cap V_j)|$.

2. $H_i^{\mathcal{T}} \cap V_j \not\subseteq Y$: In this case, we must have $|X \cap V_j| \leq (k-1)$ (otherwise $H^{\mathcal{T}} \cap V_j \subseteq Y$) and thus $|Y_j| = |Y \cap V_j| = |X \cap V_j|$. Since $H_i^{\mathcal{T}} \cap V_j \not\subseteq Y_j$ and since every node forwards in at most one stripe, *no* node of $X \cap V_j$ or $Y \cap V_j$ forwards in stripe i. Thus, $|\text{succ}_i^{\mathcal{T}}(Y \cap V_j)| = |(Y \cap V_j)| = |X \cap V_j| = |\text{succ}_i^{\mathcal{T}}(X \cap V_j)|$.

We obtain

$$a^{\mathcal{T}}(Y) = \sum_{j \in [C]} \sum_{i \in [k]} |\text{succ}_i^{\mathcal{T}}(Y \cap V_j)| \geq \sum_{j \in [C]} \sum_{i \in [k]} |\text{succ}_i^{\mathcal{T}}(X \cap V_j)| = a^{\mathcal{T}}(X). \tag{4.15}$$

□

Next, we give an upper bound on the LiSS-damage created by attacks on Cluster Topologies.

Lemma 4.1.8 ***[BSS09]***
For every attack $X \subseteq V$ on a Cluster Topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, it holds that

$$a^{\mathcal{T}}(X) \leq \sum_{i=1}^{\min(|X|, Ck)} \delta_i^{C,k}.$$

Proof. Due to Lemma 4.1.7, we can restrict the analyis to attacks $X \subseteq H^{\mathcal{T}}$. Then, it holds that $|X| \leq |H^{\mathcal{T}}| \leq Ck$.

W.l.o.g. assume that $(V_1, \ldots, V_C)$ are in order of non-increasing cardinality.

Since Equation (4.14) still applies, we inspect $a^{\mathcal{T}}(X \cap V_j)$ for each $j \in [C]$.

For $x_j := |X \cap V_j|$, we have $a^{\mathcal{T}}(X \cap V_j) = x_j|V_j| + (k - x_j)x_j$: Each node in $X \cap V_j$ is head in exactly one stripe, there having all nodes V_j as its successors. Additionally, there are $(k - x_j)$ stripes in which no node from $X \cap V_j$ has children. This leads to a LiSS-damage of x_j in each such stripe.

Thus, it holds that

$$a^{\mathcal{T}}(X \cap V_j) = \sum_{i=0}^{x_j - 1} (|V_j| + (k - 2i - 1)) = \sum_{i=0}^{x_j - 1} \delta_{Ci+j}^{C,k}. \tag{4.16}$$

Furthermore, $a^{\mathcal{T}}(X) = \sum_{j \in [C]} a^{\mathcal{T}}(X \cap V_j) = \sum_{j \in [C]} \sum_{i=0}^{x_j - 1} \delta_{Ci+j}^{C,k}$ is a sum over the damage sequence in $|X|$ distinct positions. Since the damage sequence $(\delta_i^{C,k})_{1 \leq i \leq Ck}$ is non-increasing, we obtain an upper bound by summing up its $|X|$ first elements. □

Note, that this upper bound for Cluster Topologies coincides with the lower bound Lemma 4.1.5 gave for the maximum LiSS-damage on every topology $\mathcal{T} \in \mathbb{T}(n, C, k)$. Hence, we obtain the following Theorem.

Theorem 4.1.9 ***Characterization of Optimally* LiSS*-stable Topologies***
A topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is optimally LiSS-stable if and only if it holds that

$$\forall x \in [n]\colon \max_{X \subseteq V, |X|=x} a^{\mathcal{T}}(X) = \sum_{i=1}^{\min(x, Ck)} \delta_i^{C,k}.$$

Furthermore, we have identified the Cluster Topologies as a first subset of all optimally LiSS-stable topologies. Again, this was first observed in [BSS09]. Since every class $\mathbb{T}(n, C, k)$ contains Cluster Topologies, it also contains optimally LiSS-stable topologies.

4.1.4. Properties of Optimally LiSS-Stable Topologies

Given the characterization of optimally LiSS-stable topologies in Theorem 4.1.9, we can now identify mandatory properties of such topologies. To do this, it will be convenient to give a name to attacks that bear witness to the instability of a topology.

Definition 4.1.10 ***(Minimum) Strong Attack***
An attack $X \subseteq V$ on a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is called *Strong Attack*, if it holds that

$$a^{\mathcal{T}}(X) > \sum_{i=1}^{\min(|X|, Ck)} \delta_i^{C,k}.$$

A Strong Attack X is called *Minimum Strong Attack*, if there is no Strong Attack Y with $|Y| < |X|$ and no Strong Attack Y with $|Y| = |X|$ and $a^{\mathcal{T}}(Y) > a^{\mathcal{T}}(X)$.

By definition, a topology without a (Minimum) Strong Attack must be optimally LiSS-stable. Furthermore, Minimum Strong Attacks are restricted to nodes having a high number of successors.

Lemma 4.1.11
For a Minimum Strong Attack $X \subseteq V$ on topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, it holds that

$$\forall v \in X\colon a^{\mathcal{T}}(X) - a^{\mathcal{T}}(X \setminus \{v\}) > \delta_{|X|}^{C,k}. \tag{4.17}$$

In particular, this means that

$$\forall v \in X\colon a^{\mathcal{T}}(v) > \delta_{Ck}^{C,k}. \tag{4.18}$$

Proof. We have $|X| < Ck$, since otherwise Strong Attack X would have to satisfy $a^{\mathcal{T}}(X) > \sum_{i=1}^{Ck} \delta_i^{C,k} = kn$. This is impossible because only kn source-to-peer paths exist in $\mathcal{T}$.

Assume there is a node $v \in X$ with $a^{\mathcal{T}}(X) - a^{\mathcal{T}}(X \setminus \{v\}) \leq \delta_{|X|}^{C,k}$. It holds that

$$a^{\mathcal{T}}(X \setminus \{v\}) \geq a^{\mathcal{T}}(X) - \delta_{|X|}^{C,k} > \left(\sum_{i=1}^{|X|} \delta_i^{C,k}\right) - \delta_{|X|}^{C,k} = \sum_{i=1}^{|X\setminus\{v\}|} \delta_i^{C,k}. \tag{4.19}$$

Thus, $X \setminus \{v\}$ is also a Strong Attack, contradicting that X was of minimum size.

Property (4.18) follows from Property (4.17) because $a^{\mathcal{T}}(v) \geq a^{\mathcal{T}}(X) - a^{\mathcal{T}}(X \setminus \{v\})$ and $\delta^{C,k}_{|X|} \geq \delta^{C,k}_{Ck}$. □

Using Strong Attacks, we can now identify necessary properties of optimally LiSS-stable topologies.

Corollary 4.1.12
In an optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, it holds that

$$\forall v \in V \colon a^{\mathcal{T}}(v) \leq \delta^{C,k}_1. \tag{4.20}$$

Proof. Otherwise, the set $X = \{v\}$ would be a Strong Attack. □

In every topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ with Property (4.20), we have

$$\forall i \in [k] \colon |H^{\mathcal{T}}_i| = C. \tag{4.21}$$

Otherwise, there would be a stripe $j \in [k]$ with $|H^{\mathcal{T}}_j| \leq C - 1$ (since $\sum_{i \in [k]} |H^{\mathcal{T}}_i| \leq Ck$). Since the heads $H^{\mathcal{T}}_j$ together have n successors in stripe j, there were $h \in H^{\mathcal{T}}_i$ with $|\mathrm{succ}^{\mathcal{T}}_i(h)| \geq \lceil n/(C-1) \rceil$. Additionally counting h's successors in the $(k-1)$ remaining stripes leads to $a^{\mathcal{T}}(h) \geq \lceil n/(C-1) \rceil + (k-1) > \delta^{C,k}_1$. However, this would be a contradiction to the assumption that $\mathcal{T}$ has Property (4.20).

It is also easy to see, that an optimally LiSS-stable topology in $\mathbb{T}(n, C, k)$ must have Ck *distinct* heads.

Lemma 4.1.13
An optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ has $|H^{\mathcal{T}}| = Ck$.

Proof. It holds that $|H^{\mathcal{T}}| \leq Ck$, since $\mathcal{T} \in \mathbb{T}(n, C, k)$.

If we had $|H^{\mathcal{T}}| < Ck$, the set $X = H^{\mathcal{T}}$ would be a Strong Attack: Since $\delta^{C,k}_{Ck} \geq 1$, it would hold that $a^{\mathcal{T}}(X) = kn = \sum_{i=1}^{Ck} \delta^{C,k}_i > \sum_{i=1}^{|X|} \delta^{C,k}_i$. □

The heads in each stripe need to have almost equal successor numbers.

Lemma 4.1.14
In an optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, it holds that

$$\forall i \in [k] \; \forall h \in H^{\mathcal{T}}_i \colon \lceil n/C \rceil \geq |\mathrm{succ}^{\mathcal{T}}_i(h)| \geq \lfloor n/C \rfloor. \tag{4.22}$$

Proof. The upper bound follows from Corollary 4.1.12 together with the fact that $a^{\mathcal{T}}(h) \geq |\mathrm{succ}^{\mathcal{T}}_i(h)| + (k-1)$.

For the lower bound, assume there is $h \in H^{\mathcal{T}}_i$ with $|\mathrm{succ}^{\mathcal{T}}_i(h)| < \lfloor n/C \rfloor$ and define $X := H^{\mathcal{T}}_i \setminus \{h\}$. Since $\mathcal{T}$ is stable, Equation 4.21 leads to $|X| = C - 1$.

Removing the nodes X from T_i, only paths to successors of h remain intact. This means $|\mathrm{succ}_i^{\mathcal{T}}(X)| > n - \lfloor n/C \rfloor$. Additionally, $\forall j \in [k] \setminus \{i\}\colon |\mathrm{succ}_j^{\mathcal{T}}(X)| \geq |X|$ is true by definition of the successor sets. Consequently, X is a Strong Attack:

$$a^{\mathcal{T}}(X) = \sum_{j\in[k]} |\mathrm{succ}_j^{\mathcal{T}}(X)| > n - \lfloor n/C \rfloor + (k-1)|X| \geq \sum_{j=1}^{C-1} \delta_j^{C,k}. \tag{4.23}$$

□

Heads whose successor number meets the upper bound, need to have at least one head from each stripe as their successor. Heads whose successor number meets the lower bound need to have heads from each but one other stripe among their successors.

Lemma 4.1.15
In an optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, for every stripe $i \in [k]$ every head $h \in H_i^{\mathcal{T}}$ satisfies

$$\left|\left\{j \in [k] \;\middle|\; |\mathrm{succ}_i^{\mathcal{T}}(h) \cap H_j^{\mathcal{T}}| \geq 1\right\}\right| \;\geq\; k - \left(\left\lceil \frac{n}{C} \right\rceil - |\mathrm{succ}_i^{\mathcal{T}}(h)|\right). \tag{4.24}$$

Proof. Assume there is head $h \in H_i^{\mathcal{T}}$ violating Property (4.24) and define the number $q := 1 + (\lceil n/C \rceil - |\mathrm{succ}_i^{\mathcal{T}}(h)|)$. Since, by definition, we have $h \in \mathrm{succ}_i^{\mathcal{T}}(h)$, there must be distinct $j_1, \ldots, j_q \in [k] \setminus \{i\}$ with $\mathrm{succ}_i^{\mathcal{T}}(h) \cap H_{j_r}^{\mathcal{T}} = \emptyset$ for $r \in [q]$. Since $\mathcal{T}$ is optimally LiSS-stable, Equation (4.21) applies and no node is head in two stripes.

We show that $X := \bigcup_{r\in[q]} H_{j_r}^{\mathcal{T}} \cup \{h\}$ is a Strong Attack. It holds that $|X| = Cq+1$ and $\forall r \in [q]\colon |\mathrm{succ}_{j_r}^{\mathcal{T}}(X)| = n$. Additionally, $|\mathrm{succ}_i^{\mathcal{T}}(X)| \geq Cq + (\lceil n/C \rceil - (q-1))$ follows from $|\mathrm{succ}_i^{\mathcal{T}}(X)| = |\mathrm{succ}_i^{\mathcal{T}}(X \setminus \{h\})| + |\mathrm{succ}_i^{\mathcal{T}}(h)|$. At last, paths to the nodes X are also disturbed in the remaining stripes: $\forall r \in [k] \setminus \{i, j_1, \cdots, j_q\}\colon |\mathrm{succ}_r^{\mathcal{T}}(X)| \geq Cq + 1$. Together, these observations lead to

$$a^{\mathcal{T}}(X) = \left(\sum_{r\in[q]} |\mathrm{succ}_{j_r}^{\mathcal{T}}(X)|\right) + |\mathrm{succ}_i^{\mathcal{T}}(X)| + \left(\sum_{r\in[k]\setminus\{i,j_1,\ldots,j_r\}} |\mathrm{succ}_r^{\mathcal{T}}(X)|\right) \tag{4.25}$$

$$\geq qn + \left(Cq + \left(\left\lceil \frac{n}{C} \right\rceil - (q-1)\right)\right) + (k-q-1)(Cq+1) \tag{4.26}$$

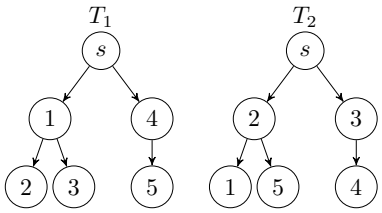
$$= qn + Cq(k-q) + \left\lceil \frac{n}{C} \right\rceil + (k-2q) \tag{4.27}$$

$$> qn + Cq(k-q) + \left\lceil \frac{n}{C} \right\rceil + (k-2q-1) = \sum_{r=1}^{Cq+1} \delta_r^{C,k}. \tag{4.28}$$

Hence, we obtain a contradiction with $\mathcal{T}$'s status of being optimally LiSS-stable. □

Note that Lemma 4.1.15 allows situations as in Figure 4.3.

Furthermore, a head from stripe i whose successor number meets the upper bound has at least one such head from each stripe $j \in [k] \setminus \{i\}$ as successor or predecessor.



(a) Topology $\mathcal{T}$

x	Optimal Attacks	$a^{\mathcal{T}}(X)$
1	$\{1\}, \{2\}$	$\sum_{i=1}^{1} \delta_i^{C,k} = 4$
2	$\{1,4\}, \{2,3\}, \{2,4\}$	$\sum_{i=1}^{2} \delta_i^{C,k} = 7$
3	$\{1,2,4\}, \{2,3,4\}$	$\sum_{i=1}^{3} \delta_i^{C,k} = 9$
4	$\{1,2,3,4\}$	$\sum_{i=1}^{4} \delta_i^{C,k} = 10$

(b) Optimal Attacks on $\mathcal{T}$

Figure 4.3.: An optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(5,2,2)$ with head 4 not supplying other heads.

Lemma 4.1.16
In an optimally LiSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, for every stripe $i \in [k]$ every head $h \in H_i^{\mathcal{T}}$ with $|\mathrm{succ}_i^{\mathcal{T}}(h)| = \lceil n/C \rceil$ satisfies

$$\forall j \in [k]\ \exists v \in H_j^{\mathcal{T}} \colon |\mathrm{succ}_j^{\mathcal{T}}(v)| = \left\lceil \frac{n}{C} \right\rceil \wedge (v \in \mathrm{succ}_i^{\mathcal{T}}(h) \vee v \in \mathrm{pred}_j^{\mathcal{T}}(h)). \tag{4.29}$$

Proof. For each stripe $j \in [k]$, define $\hat{H}_j^{\mathcal{T}} := \{v \in H_j^{\mathcal{T}} \mid |\mathrm{succ}_j^{\mathcal{T}}(v)| = \lceil \frac{n}{C} \rceil\}$.

If $n \mod C = 0$, Lemma 4.1.14 guarantees that $\forall j \in [k] \colon H_j^{\mathcal{T}} = \hat{H}_j^{\mathcal{T}}$. Consequently, Lemma 4.1.16 follows from Lemma 4.1.15.

If $n \mod C \neq 0$, assume that there are $h \in \hat{H}_i^{\mathcal{T}}$ and $j \in [k]$ with $\mathrm{succ}_i^{\mathcal{T}}(h) \cap \hat{H}_j^{\mathcal{T}} = \emptyset$ and $\mathrm{pred}_j^{\mathcal{T}}(h) \cap \hat{H}_j^{\mathcal{T}} = \emptyset$. Then the attack $X := \hat{H}_j^{\mathcal{T}} \cup \{h\}$ has cardinality $|X| \leq C$ as well as $|\mathrm{succ}_i^{\mathcal{T}}(X)| \geq |\hat{H}_j^{\mathcal{T}}| + \lceil n/C \rceil$ successors in T_i, $|\mathrm{succ}_j^{\mathcal{T}}(X)| \geq |\hat{H}_j^{\mathcal{T}}| \cdot \lceil n/C \rceil + 1$ successors in T_j, and $|\mathrm{succ}_r^{\mathcal{T}}(X)| \geq |\hat{H}_j^{\mathcal{T}}| + 1$ successors in the remaining stripes $r \in [k] \setminus \{i, j\}$.

We obtain

$$a^{\mathcal{T}}(X) = |\mathrm{succ}_i^{\mathcal{T}}(X)| + |\mathrm{succ}_j^{\mathcal{T}}(X)| + \sum_{r \in [k] \setminus \{i,j\}} |\mathrm{succ}_r^{\mathcal{T}}(X)| \tag{4.30}$$

$$\geq |\hat{H}_j^{\mathcal{T}}| + \left\lceil \frac{n}{C} \right\rceil + |\hat{H}_j^{\mathcal{T}}| \left\lceil \frac{n}{C} \right\rceil + 1 + (k-2)(|\hat{H}_j^{\mathcal{T}}| + 1) \tag{4.31}$$

$$= (|\hat{H}_j^{\mathcal{T}}| + 1) \cdot \left(\left\lceil \frac{n}{C} \right\rceil + (k-1) \right) \tag{4.32}$$

$$> |\hat{H}_j^{\mathcal{T}}| \cdot \left(\left\lceil \frac{n}{C} \right\rceil + (k-1) \right) + \left(\left\lfloor \frac{n}{C} \right\rfloor + (k-1) \right) = \sum_{r=1}^{|X|} \delta_r^{C,k}. \tag{4.33}$$

Hence, X would be a Strong Attack, which contradicts that $\mathcal{T}$ is optimally LiSS-stable. □

4.2. Rule-Based Construction of Optimally LiSS-Stable Topologies

Building on the characterization and mandatory properties we have seen in the preceding section, we now identify optimally LiSS-stable topologies that are more flexible than Cluster Topologies. In particular, we define a set of polynomial-time-checkable rules enforcing properties that are sufficient to make a topology optimally LiSS-stable. By identifying such a rule set, we give peer-to-peer topology management systems a guideline to efficiently build optimally LiSS-stable topologies without the strict limitations applying to Cluster Topologies. In this process, we aim at demanding properties that only slightly differ from the necessary properties identified in Section 4.1.4.

The rules shown in this section have been set up in cooperation with Andreas Brieg. They are published in [BBG+09].

We start with an initial set of rules which will then be modified to our needs. All rules assume, that a topology class $\mathbb{T}(n, C, k)$ is given in advance and that a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is to be constructed.

- **Not-Too-Many-Successors:**
  Ensure that $\forall v \in V : a^{\mathcal{T}}(v) \leq \delta_1^{C,k}$ and $\forall h \in H^{\mathcal{T}} : a^{\mathcal{T}}(h) \geq \delta_C^{C,k}$.
- **Head Rule 1:**
  Ensure that $\forall i, j \in [k]\ \forall h \in H_i^{\mathcal{T}} : |\mathrm{succ}_i^{\mathcal{T}}(h) \cap H_j^{\mathcal{T}}| = 1$.
- **Head Rule 2:**
  Ensure that $\forall i, j \in [k]\ \forall h \in H_i^{\mathcal{T}} : v \in \mathrm{succ}_i^{\mathcal{T}}(h) \cap H_j^{\mathcal{T}} \Rightarrow |\mathrm{succ}_i^{\mathcal{T}}(h)| = |\mathrm{succ}_j^{\mathcal{T}}(v)|$.

Note that the Not-Too-Many-Successors rule dictates the necessary properties identified in both Corollary 4.1.12 and Lemma 4.1.14 (the latter since a head of stripe i also receives all $(k-1)$ stripes $[k] \setminus \{i\}$). Head Rule 1 enforces a strengthened version of the necessary property from Lemma 4.1.15 in which the exception for heads $h \in H_i^{\mathcal{T}}$ with $|\mathrm{succ}_i^{\mathcal{T}}(h)| = \lfloor n/C \rfloor$ is eliminated. Head Rule 2 forbids successor relationships between heads with different numbers of successors in the stripe they are head of. In doing that, it guarantees that $\mathcal{T}$ has the necessary property of Lemma 4.1.16. The set of topologies in $\mathbb{T}(n, C, k)$ adhering to these rules is non-empty, since the Cluster Topologies of Section 4.1.3 are a non-empty subset.

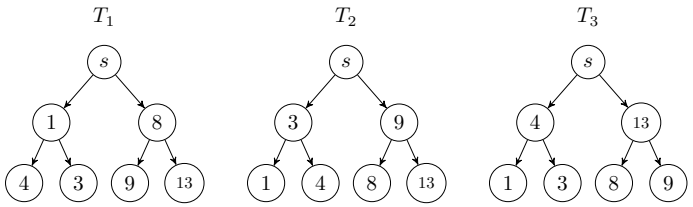
To introduce our next rule, we first have to define the concept of Head Topologies.

Definition 4.2.1 ***Head Topology $\mathcal{H}$ of $\mathcal{T}$***
For a given topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, the *head topology $\mathcal{H}$ of $\mathcal{T}$* is a distribution topology with k stripes and node set $H^{\mathcal{T}}$. Additionally, it holds that

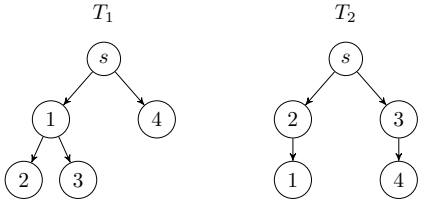
$$\forall v \in H^{\mathcal{T}}\ \forall i \in [k] : \mathrm{succ}_i^{\mathcal{H}}(v) = \mathrm{succ}_i^{\mathcal{T}}(v) \cap H^{\mathcal{T}}. \tag{4.34}$$

A topology $\mathcal{T}$ that is head topology of itself, is called *head topology*.

Hence, the head topology $\mathcal{H}$ of a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is formed from $\mathcal{T}$ by shortcutting all paths between heads having intermediate nodes only from $V \setminus H^{\mathcal{T}}$ and removing the nodes $V \setminus H^{\mathcal{T}}$ from the topology. Figure 4.4 gives two examples.



(a) Head topology of the Cluster Topology in Figure 4.2



(b) Head topology of the topology in Figure 4.3

Figure 4.4.: Examples of head topologies.

We can now state the following rule.

- **Heads-Are-Optimally-Stable:** Ensure that the head topology $\mathcal{H}$ of $\mathcal{T}$ is isomorphic to an optimally LiSS-stable topology $\mathcal{C} \in \mathbb{T}(Ck, C, k)$. Furthermore, the LiSS-stablility of $\mathcal{C}$ must be checkable in polynomial-time.

Due to Lemma 4.1.13, every optimally LiSS-stable topology $\mathcal{C} \in \mathbb{T}(Ck, C, k)$ satisfies $H^{\mathcal{C}} = V$. Thus, $\mathcal{C}$ is a head topology. Each Cluster Topology in $\mathbb{T}(Ck, C, k)$ is suitable for this rule, since its defining properties can be checked by an $O(kn)$-time tree traversal. The identification of further suitable topologies is the topic of Section 4.3.

The Heads-Are-Optimally-Stable rule enforces the necessary property of Lemma 4.1.13, since the head topology $\mathcal{H}$ of $\mathcal{T}$ has to comply with it. Thus, we have $|H^{\mathcal{T}}| = |H^{\mathcal{H}}| = Ck$.

Adherence to the Heads-Are-Optimally-Stable rule is *not a necessary condition* for optimally LiSS-stable topologies. This is demonstrated by the optimally LiSS-stable topology shown in Figure 4.3. Its head topology, shown in Figure 4.4(b), is unstable because it conflicts with the bounds on head successor numbers identified in Lemma 4.1.14.

Next, we show that topologies adhering to the above rules must be LiSS-stable against all attacks containing only heads.

Lemma 4.2.2
For a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ adhering to the rules Not-Too-Many-Successors, Head Rule 1, Head Rule 2 and Heads-Are-Optimally-Stable, it holds that

$$\forall X \subseteq H^{\mathcal{T}} \colon a^{\mathcal{T}}(X) \leq \sum_{i=1}^{|X|} \delta_i^{C,k}. \tag{4.35}$$

Proof. For $i \in [k]$, define $\breve{H}_i^{\mathcal{T}} := \{h \in H_i^{\mathcal{T}} \mid |\mathrm{succ}_i^{\mathcal{T}}(h)| = \lfloor n/C \rfloor\}$ and $\hat{H}_i^{\mathcal{T}} := H_i^{\mathcal{T}} \setminus \breve{H}_i^{\mathcal{T}}$. Due to the bounds on the successor number of heads specified by the Not-Too-Many-Successors rule, it holds that $\forall i, j \in [k]\colon |\hat{H}_i^{\mathcal{T}}| = |\hat{H}_j^{\mathcal{T}}| \wedge |\breve{H}_i^{\mathcal{T}}| = |\breve{H}_j^{\mathcal{T}}|$. With $\hat{C} := |\hat{H}_1^{\mathcal{T}}|$ and $\breve{C} := |\breve{H}_1^{\mathcal{T}}|$, we have $C = \hat{C} + \breve{C}$. Furthermore, define the sets $X_i := X \cap H_i^{\mathcal{T}}$, $\hat{X}_i := X_i \cap \hat{H}_i^{\mathcal{T}}$, and $\breve{X}_i := X_i \cap \breve{H}_i^{\mathcal{T}}$, as well as $\hat{X} := \bigcup_{i\in[k]} \hat{X}_i$ and $\breve{X} := \bigcup_{i\in[k]} \breve{X}_i$.

Now, let $\mathcal{H}$ be the head topology of $\mathcal{T}$. Due to its definition, $\mathcal{H}$ has the property that $\forall h \in H^{\mathcal{T}}\ \forall i \in [k]\colon h \in \mathrm{succ}_i^{\mathcal{H}}(X) \Leftrightarrow h \in \mathrm{succ}_i^{\mathcal{T}}(X)$. Thus, it holds that

$$a^{\mathcal{T}}(X) = \sum_{i=1}^{k} |\mathrm{succ}_i^{\mathcal{T}}(X) \cap H^{\mathcal{T}}| + \sum_{i=1}^{k} |\mathrm{succ}_i^{\mathcal{T}}(X) \setminus H^{\mathcal{T}}| \tag{4.36}$$

$$\leq a^{\mathcal{H}}(X) + \sum_{i=1}^{k} \left(|\hat{X}_i| \left\lceil \frac{n}{C} \right\rceil + |\breve{X}_i| \left\lfloor \frac{n}{C} \right\rfloor - k \cdot |X_i| \right) \tag{4.37}$$

$$\leq a^{\mathcal{H}}(X) + |\hat{X}| \left\lceil \frac{n}{C} \right\rceil + |\breve{X}| \left\lfloor \frac{n}{C} \right\rfloor - k|X|. \tag{4.38}$$

The step from Term (4.36) to (4.37) uses that the Heads-Are-Optimally-Stable rule enforces $|H^{\mathcal{T}}| = Ck$. This also means that the head sets of all stripes are disjoint. Furthermore, Head Rule 1 demands that for each stripe $i \in [k]$, each head $h \in H_i^{\mathcal{T}}$ satisfies $\forall j \in [k]\colon |\mathrm{succ}_i^{\mathcal{T}}(h) \cap H_j^{\mathcal{T}}| = 1$. Together, these properties have the effect that $|\mathrm{succ}_i^{\mathcal{T}}(h) \setminus H^{\mathcal{T}}| = |\mathrm{succ}_i^{\mathcal{T}}(h)| - k$.

Now, Head Rule 2 forbids successor relationships between the nodes $\hat{H}^{\mathcal{T}}$ and $\breve{H}^{\mathcal{T}}$. Consequently, in each stripe of $\mathcal{H}$, the node sets $\{s\} \cup \hat{H}^{\mathcal{T}}$ and $\{s\} \cup \breve{H}^{\mathcal{T}}$ induce source-rooted trees having $\hat{C}$ and $\breve{C}$ heads, respectively. Hence, over all stripes the sets induce topologies $\hat{\mathcal{H}} \in \mathbb{T}(\hat{C}k, \hat{C}, k)$ and $\breve{\mathcal{H}} \in \mathbb{T}(\breve{C}k, \breve{C}, k)$. As will be shown in Lemma 4.3.7, $\mathcal{H}$ is optimally LiSS-stable only if both $\hat{\mathcal{H}}$ and $\breve{\mathcal{H}}$ are optimally LiSS-stable. The corresponding proof does *not* depend on Lemma 4.2.2.

Consequently, we can upper-bound $a^{\mathcal{H}}(X)$ in Inequality (4.38) by the respective damage sequence sums on $\hat{\mathcal{H}}$ and $\breve{\mathcal{H}}$.

$$\begin{aligned} a^{\mathcal{T}}(X) \leq & \sum_{i=1}^{|\hat{X}|} \left(2k - \left\lfloor \frac{i-1}{\hat{C}} \right\rfloor - 1 \right) + \sum_{i=1}^{|\breve{X}|} \left(2k - \left\lfloor \frac{i-1}{\breve{C}} \right\rfloor - 1 \right) \\ & + |\hat{X}| \left\lceil \frac{n}{C} \right\rceil + |\breve{X}| \left\lfloor \frac{n}{C} \right\rfloor - k|X| \end{aligned} \tag{4.39}$$

$$= \sum_{i=1}^{|\hat{X}|} \left(\left\lceil \frac{n}{C} \right\rceil + k - \left\lfloor \frac{i-1}{\hat{C}} \right\rfloor - 1 \right) + \sum_{i=1}^{|\breve{X}|} \left(\left\lfloor \frac{n}{C} \right\rfloor + k - \left\lfloor \frac{i-1}{\breve{C}} \right\rfloor - 1 \right) \tag{4.40}$$

$$= \sum_{i=1}^{|\hat{X}|} \delta^{C,k}_{\lfloor \frac{i-1}{\hat{C}} \rfloor C + (i - \lfloor \frac{i-1}{\hat{C}} \rfloor \hat{C})} + \sum_{i=1}^{|\breve{X}|} \delta^{C,k}_{\lfloor \frac{i-1}{\breve{C}} \rfloor C + \hat{C} + (i - \lfloor \frac{i-1}{\breve{C}} \rfloor \breve{C})} \tag{4.41}$$

$$\leq \sum_{i=1}^{|X|} \delta_i^{C,k} \tag{4.42}$$

Here, the sums in Term (4.41) run over disjoint subsets of the elements of the damage sequence $(\delta_i^{C,k})_{1\leq i\leq Ck}$: For indices j summed up by the first sum, there is $q \in [0, k-1]$ such that $j \in [qC+1, qC+\hat{C}]$. For indices j in the second sum, we have $j \in [qC+\hat{C}+1, (q+1)C]$. Since the damage sequence is non-increasing, we obtain an upper bound for the Term (4.41) by summing up the first $|\hat{X}+\check{X}| = |X|$ elements of the damage sequence. This leads to Term (4.42). □

Lemma 4.2.2 shows that topologies adhering to the Heads-Are-Optimally-Stable rule, the Not-Too-Many-Successors rule, and the Head Rules 1 and 2 are optimally LiSS-stable if, for each attack size, there is a worst-case attack containing only heads.

This is exactly what the Cluster Topologies achieve. They obey all rules (their head topology is also a Cluster Topology) and have the property shown in Lemma 4.1.7.

However, instead of dictating clusters, here we choose to formulate a strengthened version of the Not-Too-Many-Successors rule.

- **Strictly-Not-Too-Many-Successors:**
  Ensure that $\forall h \in H^{\mathcal{T}}\colon \delta_1^{C,k} \geq a^{\mathcal{T}}(h) \geq \delta_C^{C,k}$ and $\forall v \in V \setminus H^{\mathcal{T}}\colon \delta_{Ck}^{C,k} \geq a^{\mathcal{T}}(v)$.

Theorem 4.2.3 ***Rules for Optimally* LiSS*-stable Topologies***
Every topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ adhering to the Strictly-Not-Too-Many-Successors rule, Head Rule 1, Head Rule 2 and the Heads-Are-Optimally-Stable rule is optimally LiSS-stable in $\mathbb{T}(n, C, k)$.

Proof. The Strictly-Not-Too-Many-Successors rule dictates $\forall v \in V \setminus H^{\mathcal{T}}\colon \delta_{Ck}^{C,k} \geq a^{\mathcal{T}}(v)$. Therefore, due to Lemma 4.1.11, every Minimum Strong Attack on $\mathcal{T}$ can contain only heads. However, the existence of a Strong Attack $X \subseteq H^{\mathcal{T}}$ is ruled out by Lemma 4.2.2. Hence, there is no Minimum Strong Attack and $\mathcal{T}$ must be optimally LiSS-stable in $\mathbb{T}(n, C, k)$. □

Note that the topologies defined by the rule set of Theorem 4.2.3 contain some, but need not contain all, Cluster Topologies from $\mathbb{T}(n, C, k)$. The primary reason for this is that, in a rule-based topology $\mathcal{T}$, a head $h \in H_i^{\mathcal{T}}$ must have at least a minimum number of children. Possible children are heads $h_j \in H^{\mathcal{T}} \setminus H_i^{\mathcal{T}}$, which must have $|\mathrm{succ}_i^{\mathcal{T}}(h_j) \leq 2|$, and non-heads $v \in V \setminus H^{\mathcal{T}}$ that have to satisfy $|\mathrm{succ}_i^{\mathcal{T}}(v)| \leq \delta_{Ck}^{C,k} - (k-1) = \left\lfloor \frac{n}{C} \right\rfloor - 2k + 2$. For $n \geq 2Ck$, the successor limit of non-heads has at least the value of the limit for heads. Hence, a head $h \in H_i^{\mathcal{T}}$ with $a^{\mathcal{T}}(h) = \delta_1^{C,k}$ must have at least

$$|\mathrm{child}_i^{\mathcal{T}}(h)| \geq \left\lceil \frac{\delta_1^{C,k} - k}{\delta_{Ck}^{C,k} - (k-1)} \right\rceil = \left\lceil \frac{\left\lceil \frac{n}{C} \right\rceil - 1}{\left\lfloor \frac{n}{C} \right\rfloor - 2k + 2} \right\rceil \tag{4.43}$$

children in T_i. For $k \geq 2$, this value is greater 1 (however, it is at most 2 for $n \geq 4Ck + 2C$). Such a lower bound on the successor number of heads is not present in the definition of Cluster Topologies.

However, the conditions on topologies stated in Theorem 4.2.3 no longer generally demand that nodes forward in at most one stripe. In the practical application of

peer-to-peer live streaming systems, we can usually assume $n \gg Ck$ and the availability of a certain amount of potent nodes that could be used as heads. Then, the majority of peers will have successor numbers far below the limits dictated by Strictly-Not-Too-Many-Successors. Thus, all these nodes obtain complete liberty in their forwarding decisions.

Furthermore, rule-based topologies promise to be more compatible with the demands posed when optimizing distribution topologies towards minimizing maximum LoSS-damage. In particular, Chapter 5 will show that the high overlap in the successor sets of heads in Cluster Topologies prevents such a minimization.

4.3. Optimally LiSS-Stable Head Topologies

Studying the rules of Theorem 4.2.3, it is quite dissatisfactory to see that, although the Heads-Are-Optimally-Stable rule demands for an optimally LiSS-stable head topology from $\mathbb{T}(Ck, C, k)$, our knowledge about these topologies is still rather limited. Indeed, the only optimally LiSS-stable head topologies known so far are the Cluster Topologies in $\mathbb{T}(Ck, C, k)$. Furthermore, each pair of such Cluster Topologies is isomorphic, so that we essentially know only a single optimally LiSS-stable head topology.

Therefore, we now study necessary and sufficient conditions leading to optimally LiSS-stable head topologies. The obtained results are published in [GFS11].

The analysis in this section is based on the concept of *dependency graphs*, which will be introduced in Subsection 4.3.1. The following Subsection 4.3.2 highlights special stability properties of head topologies having unconnected dependency graphs. Finally, we engage in a general study of the dependency graphs of optimally LiSS-stable head topologies in Subsection 4.3.3.

4.3.1. A Specialized Stability Characterization

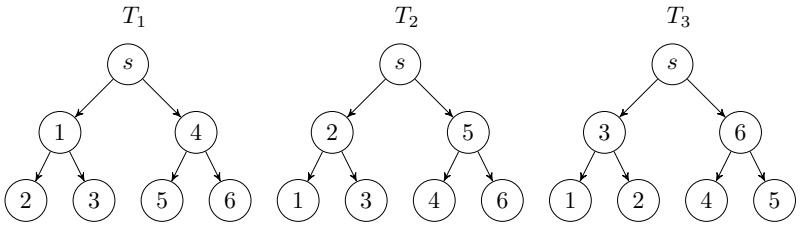
For the following analysis, we transform the damage-based characterization of optimally LiSS-stable topologies in Theorem 4.1.9 into a graph-based characterization that is specialized for head topologies.

The starting point for this step is the following concept.

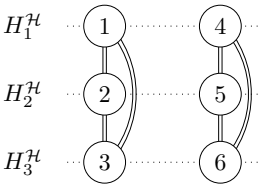
Definition 4.3.1 ***Dependency Graph of Head Topology $\mathcal{H}$***
Let $\mathcal{H}$ be a head topology with peers V. The *dependency graph* $D(\mathcal{H}) = (V, A)$ *of head topology* $\mathcal{H}$ is a loopless, undirected multigraph, such that A contains each edge $\{u, v\}$ with $u, v \in V, u \neq v$ exactly $|\{i \in [k] \mid u \in \text{succ}_i^{\mathcal{H}}(v) \vee v \in \text{succ}_i^{\mathcal{H}}(u)\}|$ times.

Figure 4.5 gives an example of such a dependency graph. If possible, we will generally use the convention to draw dependency graphs such that heads from the same head set $H_i^{\mathcal{H}}$ are aligned horizontally.

Based on our findings in Section 4.1.4, we already know the following necessary properties of optimally LiSS-stable topologies in $\mathbb{T}(Ck, C, k)$.



(a) A head topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ that is also a Cluster Topology.



(b) Corresponding dependency graph.

Figure 4.5.: A clustered head topology and its dependency graph.

Corollary 4.3.2
An optimally LiSS-stable topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ has the following properties:

1. $|H^{\mathcal{H}}| = Ck$
2. $\forall i \in [k]\colon |H_i^{\mathcal{H}}| = C$
3. $\forall i, j \in [k]\ h \in H_i^{\mathcal{H}}\colon |\mathrm{succ}_i^{\mathcal{H}}(h) \cap H_j^{\mathcal{H}}| = 1$
4. $\forall h \in H^{\mathcal{H}}\ !\exists i \in [k]\colon |\mathrm{succ}_i^{\mathcal{H}\rightarrow}(h) > 0|$

Proof. Property 1 follows from Lemma 4.1.13, Property 2 from Corollary 4.1.12, Property 3 from Lemma 4.1.15 and Property 4 from the combination of the Properties 1 and 2, Corollary 4.1.12 and Lemma 4.1.14. □

The dependency graph $D(\mathcal{H})$ of a topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ with all properties from Corollary 4.3.2 has the following characteristic features:

- $D(\mathcal{H})$ is k-partite with parts $H_i^{\mathcal{T}}$ for $i \in [k]$: In a distribution topology, heads of the same stripe i have no successor relationships in T_i. Additionally, Property 4 forbids to have successors in other stripes.

- For each $i \in [k]$ and each head $h \in H_i^{\mathcal{H}}$ the multiplicity (cmp. Section 2.2.1) of h and each set $H_j^{\mathcal{H}}$ with $i \neq j$ is $m_{D(\mathcal{H})}(\{h\}, H_j^{\mathcal{H}}) = 2$: Due to Property 1, it holds that $H_i^{\mathcal{H}} \cap H_j^{\mathcal{H}} = \emptyset$. So, due to the Property 3, h has exactly one head from $H_j^{\mathcal{H}}$ among its successors in stripe i and there must be exactly one predecessor of h

from $H_j^{\mathcal{T}}$. Finally, h and the heads from $H_j^{\mathcal{H}}$ have successors besides themself only in stripe i resp. j (Property 4).

- $D(\mathcal{H})$ is $2(k-1)$-regular: This follows from the above observations.

Property 4 also causes that $\forall i \in [k]\colon h \in V \setminus H_i^{\mathcal{H}} \Leftrightarrow \mathrm{succ}_i^{\mathcal{H}}(h) = \{h\}$ and

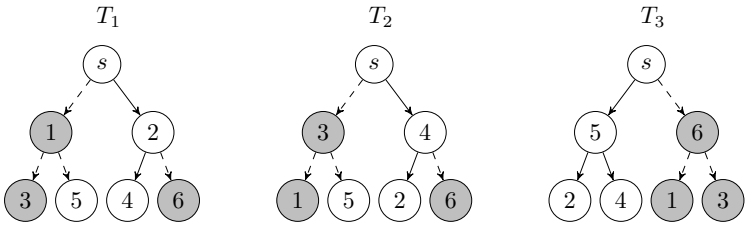
$$\forall i,j \in [k]\ \forall h \in H_i^{\mathcal{H}}\colon |\mathrm{pred}_j^{\mathcal{H}}(h) \setminus \{s,h\}| = \begin{cases} 0 & , \text{ if } j = i \\ 1 & , \text{ otherwise.} \end{cases} \tag{4.44}$$

Since $\forall i \in [k]\ \forall X \subseteq V\colon X \subseteq \mathrm{succ}_i^{\mathcal{H}}(X)$ is true due to the definition of successor sets, we can then express the LISS-damage of an attack X on $\mathcal{H}$ in the following way (cmp. Section 2.2.1 for the definition of $e_{D(\mathcal{H})}(X)$).

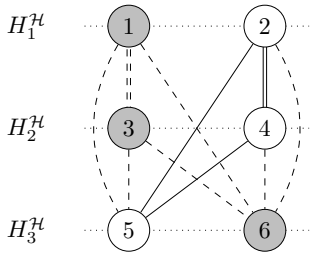
$$\begin{aligned}
a^{\mathcal{H}}(X) &= \sum_{i\in[k]} |\mathrm{succ}_i^{\mathcal{H}}(X)| && (4.45)\\
&= \sum_{i\in[k]} |X \cap H_i^{\mathcal{H}}| + (|X \setminus H_i^{\mathcal{H}}| + |\mathrm{succ}_i^{\mathcal{H}}(X) \setminus X|) && (4.46)\\
&= |X| + \sum_{i\in[k]} \sum_{v\in X} |\{v\} \setminus H_i^{\mathcal{H}}| + |\mathrm{succ}_i^{\mathcal{H}}(v) \setminus \{v\}| - |(\mathrm{succ}_i^{\mathcal{H}}(v) \setminus \{v\}) \cap X| && (4.47)\\
&= |X| + \sum_{i\in[k]} \sum_{v\in X} |\mathrm{pred}_i^{\mathcal{H}}(v) \setminus \{s,v\}| + |\mathrm{succ}_i^{\mathcal{H}}(v) \setminus \{v\}| \\
&\qquad - \sum_{i\in[k]} \sum_{v\in X} |(\mathrm{succ}_i^{\mathcal{H}}(v) \setminus \{v\}) \cap X| && (4.48)\\
&= |X| + \sum_{v\in X} \sum_{i\in k} |\{u \in V \setminus \{v\} \mid v \in \mathrm{succ}_i^{\mathcal{H}}(u) \vee u \in \mathrm{succ}_i^{\mathcal{H}}(v)\}| \\
&\qquad - \sum_{v\in X} \sum_{i\in[k]} |\{u \in X \setminus \{v\} \mid u \in \mathrm{succ}_i^{\mathcal{H}}(v)\}| && (4.49)\\
&= |X| + \sum_{v\in X} m_{D(\mathcal{H})}(v) - \frac{1}{2} \sum_{v\in X} m_{D(\mathcal{H})[X]}(v) && (4.50)\\
&= |X| + \left(\sum_{v\in X} m_{D(\mathcal{H})}(v) \right) - e_{D(\mathcal{H})}(X) && (4.51)
\end{aligned}$$

Hence, $a^{\mathcal{H}}(X)$ equals the value $|X|$ plus the multiplicity sum over all nodes X minus the number of edges incident to *two* nodes from X. Furthermore, this equals $|X|$ plus the number of edges of $D(\mathcal{H})$ that are incident to X. See Figure 4.6 for an example of this equality.

Next, we establish the following definition.



(a) Attacked head topology $\mathcal{H}$.



(b) Dependency graph $D(\mathcal{H})$

Figure 4.6.: Attack $X = \{1, 3, 6\}$ in a head topology $\mathcal{H}$ and its dependency graph $D(\mathcal{H})$ (dashed connections lost).

Definition 4.3.3 ***The Edge Sequence*** σ_i^C
For $C, k \in \mathbb{N}$, the elements of the *edge sequence* $(\sigma_i^C)_{1 \le i \le Ck}$ are defined by

$$\sigma_i^C := 2 \left\lfloor \frac{i-1}{C} \right\rfloor .$$

For $x \in [0, k-1]$ and $y \in [C]$, we can give the following closed form expression for sums over the first $Cx + y$ elements of the edge sequence.

$$\sum_{i=1}^{Cx+y} \sigma_i^C = \sum_{i=1}^{Cx} 2 \left\lfloor \frac{i-1}{C} \right\rfloor + \sum_{i=Cx+1}^{Cx+y} 2 \left\lfloor \frac{i-1}{C} \right\rfloor \tag{4.52}$$

$$= 2C \left(\sum_{j=1}^{x} (j-1) \right) + y \cdot 2x \tag{4.53}$$

$$= Cx(x-1) + 2xy \tag{4.54}$$

The edge sequence plays an important role in the characterization of optimally LiSS-stable head topologies using dependendy graphs.

Lemma 4.3.4 ***Characterization of Optimally* LiSS*-stable Head Topologies***
A head topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ is optimally LiSS-stable if and only if it has the properties given in Corollary 4.3.2 and for each $X \subseteq V$ with $|X| = Cx + y$ and $x \in [0, k-1]$, $y \in [C]$, it holds that

$$e_{D(\mathcal{H})}(X) \geq \sum_{i=1}^{|X|} \sigma_i^C = Cx(x-1) + 2xy.$$

Proof. The properties given in Corollary 4.3.2 were shown to be necessary.

Since $n = Ck$, it holds that $\delta_1^{C,k} = 2k - 1$ and $\sigma_i^C = \delta_1^{C,k} - \delta_i^{C,k}$. Furthermore, $D(\mathcal{H})$ must be $2(k-1)$-regular. Using Equation (4.51), we obtain

$$a^{\mathcal{H}}(X) = |X| + \left(\sum_{v \in X} m_{D(\mathcal{H})}(v) \right) - e_{D(\mathcal{H})}(X) \tag{4.55}$$

$$\leq |X| + |X|(2k-2) - \sum_{i=1}^{|X|} \sigma_i^C \tag{4.56}$$

$$= |X| \cdot \delta_1^{C,k} - \sum_{i=1}^{|X|} (\delta_1^{C,k} - \delta_i^{C,k}) \tag{4.57}$$

$$= \sum_{i=1}^{|X|} \delta_i^{C,k}. \tag{4.58}$$

Consequently, Lemma 4.3.4 follows from Theorem 4.1.9. □

We see, that a Strong Attack X with $|X| = Cx + y$ on a topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ with the properties given in Corollary 4.3.2 will *induce less than* $Cx(x-1) + 2xy$ *edges in* $D(\mathcal{H})$. Hence, the dependency graphs of optimally LiSS-stable head topologies have to originate from a (multi-)graph family *without sparse induced subgraphs*. In the following, we will aim to identify such families.

4.3.2. The Case of Unconnected Dependency Graphs

At first, the LiSS-stability of head topologies with unconnected dependency graphs is led back to the LiSS-stability of those with connected dependency graphs. This will allow us to strengthen results in the following Subsection 4.3.3.

Definition 4.3.5 ***Subtopologies of* $\mathcal{H}$**
Let $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ be a head topology, such that $D(\mathcal{H})$ has r connected components. Let $V_1, \ldots, V_r$ be the node sets of these components.

For each $i \in [r]$, the following tupel is called *subtopology of* $\mathcal{H}$:

$$\mathcal{H}_i = (T_1[\{s\} \cup V_i], \ldots, T_k[\{s\} \cup V_i])$$

Lemma 4.3.6
Let $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ be a head topology with the properties given in Corollary 4.3.2 and subtopologies $\mathcal{H}_1, \dots, \mathcal{H}_r$.

For each $i \in [r]$, $\mathcal{H}_i$ is a head topology from $\mathbb{T}(C_i k, C_i, k)$, where $C_i \in [C]$. Furthermore, $\mathcal{H}_i$ has the properties given in Corollary 4.3.2.

Proof. $D(\mathcal{H})$ is k-partite with parts $H_1^{\mathcal{H}}, \dots, H_k^{\mathcal{H}}$ and each node has 2 edges into each part besides its own. Therefore, each connected component i must contain the same number of heads from every stripe: $\forall a, b \in [k]\colon |V_i \cap H_a^{\mathcal{H}}| = |V_i \cap H_b^{\mathcal{H}}|$. Otherwise, there would be stripes $a, b \in [k]$ with $|V_i \cap H_a^{\mathcal{H}}| < |V_i \cap H_b^{\mathcal{H}}|$, so that $m_{D(\mathcal{H})}(V_i \cap H_a^{\mathcal{H}}, V_i \cap H_b^{\mathcal{H}}) = 2|V_i \cap H_a^{\mathcal{H}}| < 2|V_i \cap H_b^{\mathcal{H}}| = m_{D(\mathcal{H})}(V_i \cap H_b^{\mathcal{H}}, V_i \cap H_a^{\mathcal{H}})$. However, since both sets are disjoint, their multiplicity must be symmetric and we obtain a contradiction.

Since $D(\mathcal{H})$ contains edges for each successor relation between nodes $u, v \in V$, the induced subgraph $T_a[\{s\} \cup V_i]$ is connected for each node set V_i and stripe T_a.

Furthermore, each $\mathcal{H}_i$ inherits its successor relationships from $\mathcal{H}$. Hence, it is a head topology and all remaining properties given in Corollary 4.3.2 apply. □

Lemma 4.3.7
A head topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ with the properties given in Corollary 4.3.2 and subtopologies $\mathcal{H}_1, \dots, \mathcal{H}_r$ is optimally LiSS-stable in $\mathbb{T}(Ck, C, k)$ if and only if, for each $i \in [r]$, the subtopology $\mathcal{H}_i$ is optimally LiSS-stable in $\mathbb{T}(C_i k, C_i, k)$.

Proof. "Only-If": Assume there exists a Strong Attack X_a on subtopology $\mathcal{H}_a$ such that $|X_a| = C_a x_a + y_a$ and $x_a \in [0, k-1], y_a \in [C_a]$.

For each $j \in [r] \setminus \{a\}$, define sets $X_j := \bigcup_{s=1}^{x_a} V_j \cap H_s^{\mathcal{H}}$ formed by the heads of the first x_a stripes in $\mathcal{H}_j$, respectively. Due to Lemma 4.3.6, we have a multiplicity of $2C_j$ between the heads of each stripe pair in $\mathcal{H}_j$. Heads of the same stripe do not have common edges. This leads to $e_{D(\mathcal{H})}(X_j) = 2C_j\binom{x_a}{2} = C_j x_a(x_a - 1)$.

Since the peer sets $V_1, \dots, V_r$ are a partition of V, it holds that $\sum_{i=1}^r C_i = C$ and the set $X := \bigcup_{i=1}^r X_i$ has $(\sum_{i=1}^r C_i)x_a + y_a$ nodes. Since $D(\mathcal{H})$ contains no edges between nodes of different subtopologies, X induces $e_{D(\mathcal{H})}(X) = \sum_{i=1}^r e_{D(\mathcal{H})}(X_i)$ edges. Our assumption $e_{D(\mathcal{H})}(X_a) < C_a x_a(x_a - 1) + 2x_a y_a$ then leads to

$$e_{D(\mathcal{H})}(X) < (\sum_{i=1}^{r} C_i)x_a(x_a - 1) + 2x_a y_a = \sum_{i=1}^{|X|} \sigma_i^C. \tag{4.59}$$

Hence, X is a Strong Attack and $\mathcal{H}$ is not optimally LiSS-stable.

"If": Assume that all subtopologies are optimally LiSS-stable. Let X be an arbitrary attack on $\mathcal{H}$ and define $X_i := X \cap V_i$ for each $i \in [r]$.

It holds that $\sum_{i=1} C_i = C$, $\sum_{i=1} |X_i| = |X|$, and $\forall i \in [r]\colon e_{D(\mathcal{H}_i)}(X_i) \geq \sum_{a=1}^{|X_i|} \sigma_a^{C_i}$.

Furthermore, the edge sequence $(\sigma_a^C)_{1 \leq a \leq Ck}$ has the following property:

$$\forall i \in [r], p \in [0, k-1], q \in [C_i]\colon \sigma^C_{pC+q+\sum_{a=1}^{i-1} C_a} = \sigma^{C_i}_{pC_i+q} = 2p. \tag{4.60}$$

For $i, j \in [r], i \neq j$, these equivalencies map the elements of $(\sigma_a^{C_i})_{1 \leq a \leq C_i k}$ and $(\sigma_a^{C_j})_{1 \leq a \leq C_j k}$ to disjoint index regions of $(\sigma_a^C)_{1 \leq a \leq Ck}$. Since subtopologies represent connected components in $D(\mathcal{H})$ and since the edge sequence is non-decreasing, we can write

$$e_{D(\mathcal{H})}(X) = \sum_{i \in [r]} e_{D(\mathcal{H}_i)}(X_i) \geq \sum_{i \in [r]} \sum_{a=1}^{|X_i|} \sigma_a^{C_i} \geq \sum_{a=1}^{|X|} \sigma_a^C. \tag{4.61}$$

Hence, X is not a Strong Attack. Since it was chosen arbitrarily, the same applies to all attacks on $\mathcal{H}$. □

4.3.3. Dependency Graphs of Optimally LiSS-Stable Head Topologies

Now, we study dependency graphs of optimally LiSS-stable head topologies.

Given a head topology $\mathcal{H}$, we will often be interested in the *neighborhood* of a peer $v \in V$ in $D(\mathcal{H}) = (V, A)$. It is defined as

$$N(v) := \{u \in V \mid \{u, v\} \in A\} \quad \text{and} \quad N_i(v) := N(v) \cap H_i^{\mathcal{H}}.$$

Multigraphs in which the neighborhood of each node induces a quite dense subgraph are the claw-free graphs.

Definition 4.3.8 ***Claw-Free Graph [BLS99]***
A multigraph $G = (V, E)$ is called *claw-free*, if for each triple of distinct nodes $v_1, v_2, v_3 \in V$ with $v_1, v_2, v_2 \in N(v)$ for some $v \in V$, it holds that $e_G(\{v_1, v_2, v_3\}) \geq 1$.

Claw-free graphs obtained their name from the absence of the $K_{1,3}$, the *claw graph*, as an induced subgraph. A direct consequence of this definition is that for nodes $v \in V$ and $u \in N(v)$, the nodes $N(v) \setminus N(u)$ have to induce a clique. Otherwise v, u and two nodes from $N(v) \setminus N(u)$ without a common edge would induce a claw.

The claw-freeness of a given graph $G = (V, E)$ can naïvely be determined in $O(|V|^4)$ by testing for each combination of 4 nodes whether they induce a claw.

We show that claw-freeness is a necessary condition on the dependency graphs of optimally LiSS-stable head topologies.

Definition 4.3.9 ***Head Stability Requirements***
Let $\mathcal{H}$ be a head topology from $\mathbb{T}(Ck, C, k)$. The following requirements are called *Stability Requirements*.

1. $D(\mathcal{H})$ is claw-free.
2. For *distinct* stripes $a, i, j \in [k]$, it holds that

$$\forall v \in H_a^{\mathcal{H}} \; \forall u \in N_i(v) \colon m_{D(\mathcal{H})}(\{u\}, N_j(v)) \geq \frac{2}{|N_i(v)|}.$$

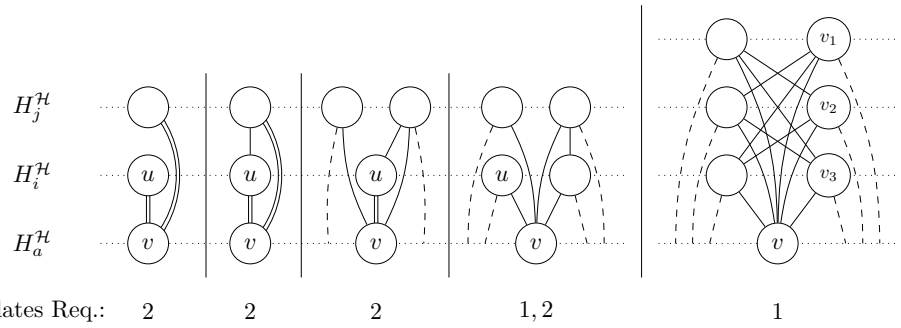


Figure 4.7.: Neighborhoods violating Stability Requirements of Definition 4.3.9 (edges to unknown nodes from $H_a^{\mathcal{H}} \setminus \{v\}$ dashed).

Theorem 4.3.10
Every optimally LiSS-stable head topology satisfies the Stability Requirements.

Proof. Assume that head topology $\mathcal{H}$ is optimally LiSS-stable and violates Requirement 2. For $v \in H_a^{\mathcal{H}}$ let $u \in N_i(v)$ be the node without the required multiplicity to $N_j(v)$. $\mathcal{H}$ must have all properties given in Corollary 4.3.2 and thus, in $D(\mathcal{H})$, each head has multiplicity two to the head sets of other stripes. Therefore, for all $q \in [k] \setminus \{a\}$ and $w \in N_q(v)$, we must have $m_{D(\mathcal{H})}(w, v) = m_{D(\mathcal{H})}(v, w) = 2/|N_q(v)|$. This also leads to $m_{D(\mathcal{H})}(w, H_a^{\mathcal{H}} \setminus \{v\}) = 2 - 2/|N_q(v)|$. Since $H_a^{\mathcal{H}}$ and $N_j(v)$ both are independent sets in $D(\mathcal{H})$, the node set $X := \{u\} \cup N_j(v) \cup H_a^{\mathcal{H}} \setminus \{v\}$ induces

$$e_{D(\mathcal{H})}(X) = m_{D(\mathcal{H})}(\{u\}, H_a^{\mathcal{H}} \setminus \{v\}) + m_{D(\mathcal{H})}(\{u\}, N_j(v)) + m_{D(\mathcal{H})}(N_j(v), H_a^{\mathcal{H}} \setminus \{v\}) \tag{4.62}$$

$$< \left(2 - \frac{2}{|N_i(v)|}\right) + \frac{2}{|N_i(v)|} + |N_j(v)| \cdot \left(2 - \frac{2}{|N_j(v)|}\right) \tag{4.63}$$

$$= 2 \cdot |N_j(v)| = \sum_{r=1}^{C+|N_j(v)|} \sigma_r^C \tag{4.64}$$

edges using $C + |N_j(v)|$ nodes. Hence, X is a Strong Attack.

Now assume $\mathcal{H}$ violates Requirement 1 and let $v_1, v_2, v_3, v \in V$ with $v_1, v_2, v_3 \in N(v)$ induce a claw in $D(\mathcal{H})$. There is $a \in [k]$ with $v \in H_a^{\mathcal{H}}$. Now, $X := \{v_1, v_2, v_3\} \cup H_a^{\mathcal{H}} \setminus \{v\}$ is a Strong Attack of $C + 2$ nodes inducing at most 3 edges: Both $\{v_1, v_2, v_3\}$ and $H_a^{\mathcal{H}}$ are independent sets and each $v_i \in \{v_1, v_2, v_3\}$ can have at most one other edge into $H_a^{\mathcal{H}}$ besides its edge(s) to v.

The identification of a Strong Attack in both cases contradicts the assumption that $\mathcal{H}$ was optimally LiSS-stable. □

Note that Requirement 2 strengthens Requirement 1 in neighborhoods containing parallel edges. This has interesting consequences.

Corollary 4.3.11
In the dependency graph $D(\mathcal{H}) = (V, A)$ of an optimally LiSS-stable head topology $\mathcal{H}$, the set of parallel edges induces a subgraph consisting only of cliques. Additionally, for each pair $u, v \in V$ with $m_{D(\mathcal{H})}(u, v) = 2$, it holds that $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

Proof. Otherwise, Requirement 2 of Theorem 4.3.10 is violated. In particular, there are distinct $a, i, j \in [k]$, such that $v \in H_a^{\mathcal{H}}$, $u \in H_i^{\mathcal{H}}$, and $N_j(v) \neq N_j(u)$. Assuming that $D(\mathcal{H})$ is claw-free, neighborhoods isomorphic to one of the three left-most neighborhoods in Figure 4.7 could be found. □

If the dependency graph of an optimally LiSS-stable head topology in $\mathbb{T}(Ck, C, k)$ consists *only* of cliques with parallel edges, it must contain C cliques of k nodes (due to the multiplicity constraints). These are exactly the dependency graphs of Cluster Topologies in $\mathbb{T}(Ck, C, k)$ (e.g., see Figure 4.5), since they have only tit-for-tat successor relationships between their heads. Thus, we have shown that Cluster Topologies are the only optimally LiSS-stable head topologies with only parallel edges in their dependency graph.

In the next step, we identify a property of dependency graphs that is sufficient to guarantee optimal LiSS-stability. For this, we have to introduce yet another graph class: the line graphs. From the possible characterizations, we choose a more uncommon one.

Definition 4.3.12 ***Line Graph [MM99]***
A multigraph $G = (V, E)$ is called *line graph*, if E can be partitioned into sets $E_1, \ldots, E_r$ such that for each $i \in [r]$ the set E_i induces a simple clique in G and each $v \in V$ is member in exactly two such cliques.

Figure 4.8 shows a dependency graph that is a line graph. The question whether a given graph $G = (V, E)$ is a line graph can be determined in time $O(|V| + |E|)$ [Rou73].

Theorem 4.3.13 ***Line Graph Criterion***
A head topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ is optimally LiSS-stable if it has the properties listed in Corollary 4.3.2 and $D(\mathcal{H}) = (V, A)$ is a line graph.

Proof. Let $X \subseteq V$ with $|X| = Cx + y$ (s.t. $x \in [0, k-1], y \in [C]$) be an arbitrary attack on $\mathcal{H}$ and let $A_1, \ldots, A_r$ be the line graph's characteristic partition of the edge set A. Slightly abusing notation, we will also use $A_1, \ldots, A_r$ to denote the (complete) subgraphs of $D(\mathcal{H})$ that are induced by the respective edge sets. For $i \in [r]$, the nodes from X occuring in A_i will be denoted as X_i.

Since $D(\mathcal{H})$ is k-partite, $2(k-1)$-regular, and each node is incident to exactly two cliques, each of the cliques $A_1, \ldots, A_r$ has to contain k nodes and there must be exactly $r = 2C$ such cliques. Since $A_1, \ldots, A_r$ is a partition of the edge set, it holds that

$$e_{D(\mathcal{H})}(X) = \sum_{i=1}^{2C} e_{A_i}(X_i). \tag{4.65}$$

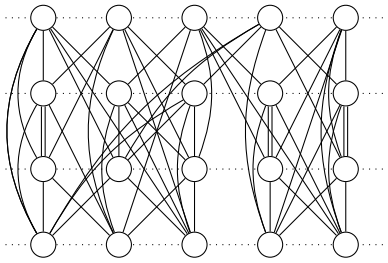


Figure 4.8.: A dependency graph that is a line graph.

For $i \in [2C]$, set $x_i := |X_i|$. Since A_i is a simple clique, the x_i nodes induce $\binom{x_i}{2}$ edges in A_i. Furthermore, each node $v \in V$ is member in exactly two cliques. This leads to $\sum_{i=1}^{2C} x_i = 2|X| = 2(Cx + y)$.

Due to Lemma A.0.2, the number of overall induced edges will be minimized if $\forall i \in [2C] \colon x_i \in \left[\left\lfloor \frac{2(Cx+y)}{2C} \right\rfloor, \left\lceil \frac{2(Cx+y)}{2C} \right\rceil\right] = [x, x+1]$. In such a setting there are $2y$ cliques with an index i such that $x_i = x + 1$. Thus, it holds that

$$e_{D(\mathcal{H})}(X) = \sum_{i=1}^{2C} \frac{x_i(x_i - 1)}{2} \tag{4.66}$$

$$\geq (2C - 2y)\frac{x(x-1)}{2} + 2y\frac{(x+1)x}{2} \tag{4.67}$$

$$= Cx(x-1) + 2xy. \tag{4.68}$$

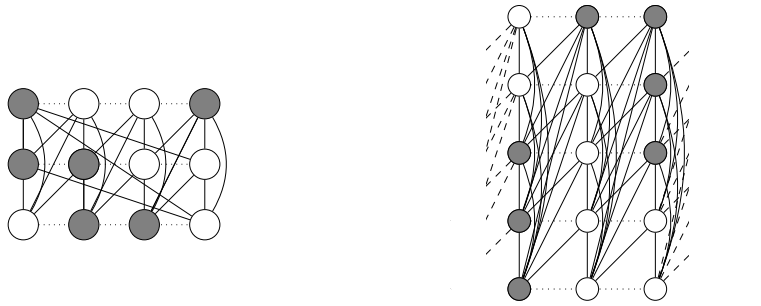
Consequently, $\mathcal{H}$ meets the characterization of Lemma 4.3.4. □

Theorem 4.3.13 has interesting implications for Strong Attacks on head topologies.

Corollary 4.3.14
A Strong Attack X on a head topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ with the properties given in Corollary 4.3.2 contains heads of at least 3 stripes of $\mathcal{H}$.

Proof. If $X \subseteq H_a^{\mathcal{H}} \cup H_b^{\mathcal{H}}$ for $a, b \in [k]$, then it holds that $e_{D(\mathcal{H})}(X) = e_{D(\mathcal{H})[H_a^{\mathcal{H}} \cup H_b^{\mathcal{H}}]}(X)$. However, due to the properties given in Corollary 4.3.2, $D(\mathcal{H})[H_a^{\mathcal{H}} \cup H_b^{\mathcal{H}}]$ has $2C$ nodes, $2C$ edges, and is 2-regular. Since every edge is a 2-clique, this graph is a line graph. Furthermore, it is a valid dependency graph for a head topology from $\mathbb{T}(2C, C, 2)$. Hence, as shown in the proof of Theorem 4.3.13, X induces more than $\sum_{i=1}^{|X|} \sigma_i^C$ edges and cannot be a Strong Attack. □

The class of optimally LISS-stable head topologies described by the Line Graph Criterion contains the Cluster Topologies (which have dependency graphs consisting of C cliques with *parallel* edges). Membership can be checked in linear time and appropriate dependency graphs can easily be constructed by packing $2C$ simple k-cliques into an edgeless multigraph while maintaining the multiplicity limitations and k-partiteness.



(a) 3 induced edges, $|X| = C + 2$ (b) 9 induced edges, $|X| = 2C + 1$

Figure 4.9.: Strong Attacks in dependency graphs of head topologies conforming to the Stability Requirements (dashed edges wrap around, attacked nodes gray).

However, the topologies conforming to the Stability Requirements but that do not meet the Line Graph Criterion still form a *gap* containing head topologies of unknown LiSS-stability status. As can be seen in the Figures 4.9 and 4.10, this gap contains both optimally LiSS-stable and unstable head topologies. The LiSS-stability of the depicted examples can be verified by the forthcoming results.

To obtain insights about the LiSS-stability of head topologies in this gap, we investigate the existence conditions of Minimum Strong Attacks.

Lemma 4.3.15 ***Properties of Minimum Strong Attacks***
Let $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ meet the Stability Requirements and have the properties given in Corollary 4.3.2. A Minimum Strong Attack X with $|X| = Cx + y$ (s.t. $x \in [0, k-1], y \in [C]$) has the following properties:

1. $C + 2 \leq Cx + y \leq \frac{Ck}{2}$
2. $y \leq \frac{Cx+2}{2x+1}$
3. $e_{D(\mathcal{H})}(X) = Cx(x-1) + 2xy - 1$
4. $\max_{v \in X} m_{D(\mathcal{H})}(v, X) = 2x - 1$

Proof. First, we prove the lower bound in Property 1. We must have $x \geq 1$, since no attack can induce less than $C \cdot 0 \cdot (0-1) + 2y \cdot 0 = 0$ edges. Hence, assume that $|X| = C + 1$. Then, the Strong Attack X induces $\epsilon \in \{0, 1\}$ edges in $D(\mathcal{H})$. Let X_ϵ be the (0 or 2) nodes of X inducing an edge.

We study the multiplicity between $V \setminus X$ and X in $D(\mathcal{H})$. Figure 4.11(a) shows the possible neighborhoods of each $v \in V \setminus X$ with $|X_\epsilon \cap N(v)| < 2$ in X. It has to hold that $|N(v) \cap X| \leq 2$, since otherwise we had a claw in $D(\mathcal{H})$. In particular, it holds that $m_{D(\mathcal{H})}(\{v\}, X) \leq 2$, because if v has parallel edges to a node $u \in X$ then u has edges to all other neighbors of v, due to Requirement 2 of the Stability Requirements.

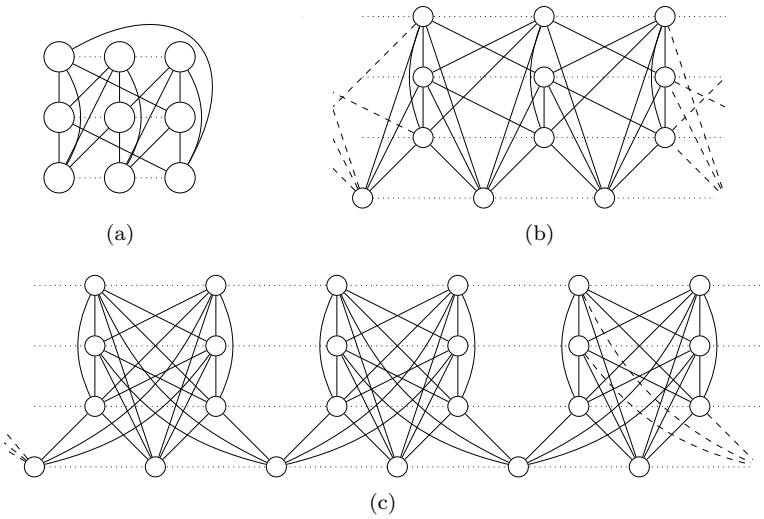
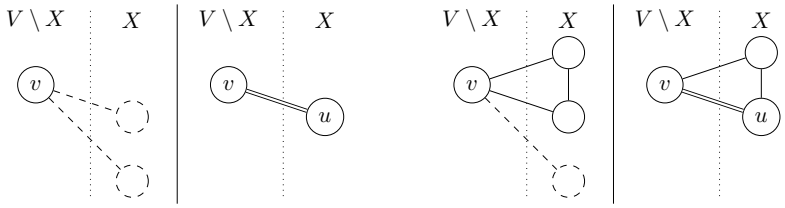


Figure 4.10.: Dependency graphs of optimally LiSS-stable head topologies violating the Line Graph Criterion (dashed edges wrap around).

By assumption, $N(v) \cap X$ is an independent set. Hence, if v has parallel edges, it holds that $N(v) \cap X = \{u\}$ and $m_{D(\mathcal{H})}(\{v\}, X) = 2$.

Due to analogue arguments, each $v \in V \setminus X$ with $|X_\epsilon \cap N(v)| = 2$ has $|N(v) \cap X| \leq 3$ and $m_{D(\mathcal{H})}(\{v\}, X) \leq 3$. Figure 4.11(b) shows the occuring neighborhoods. If such nodes exist, they are neighbors to *both* nodes X_ϵ. Hence, they must be heads of different stripes than the nodes X_ϵ and there can be at most $2(k-2)$ of them. With these upper bounds on the multiplicity of individual nodes $v \in V \setminus X$ to X, we obtain an upper bound for the whole set:

$$m_{D(\mathcal{H})}(V \setminus X, X) \leq 2|V \setminus X| + \epsilon(2(k-2)) = 2(Ck - (C+1)) + \epsilon(2(k-2)). \quad (4.69)$$



(a) The case $|N(v) \cap X_e| < 2$. (b) The case $|N(v) \cap X_\epsilon| = 2$.

Figure 4.11.: Possible neighborhoods of $v \in V \setminus X$ in X in the proof of Lemma 4.3.15. Dashed objects could, but need not exist.

Since $D(\mathcal{H})$ is $2(k-1)$-regular, the number of edges leaving the set X is

$$m_{D(\mathcal{H})}(X, V \setminus X) = (C+1)2(k-1) - 2\epsilon. \tag{4.70}$$

This implies $m_{D(\mathcal{H})}(V \setminus X, X) < m_{D(\mathcal{H})}(X, V \setminus X)$, because $k \geq 1$ and $\epsilon \in \{0, 1\}$. However, this is impossible since X and $V \setminus X$ are disjoint. Hence, a Strong Attack X with $|X| < C + 2$ does not exist.

Next, we prove that $D(\mathcal{H})[X]$ contains a node of multiplicity at least $2x - 1$. From $|X| \geq C + 2$ follows $e_{D(\mathcal{H})}(X) \geq 1$. Otherwise, X would contain an independent set of cardinality $C + 1$, which would be a smaller Strong Attack than X. Since X is a Minimum Strong Attack, this is impossible. Let $v \in X$ be a node incident to an edge in $D(\mathcal{H})[X]$. The subset $X' := X \setminus \{v\}$ has to satisfy $e_{D(\mathcal{H})}(X') \geq Cx(x-1) + 2x(y-1)$ (also for $y = 1$, since $Cx(x-1) = C(x-1)(x-2) + 2C(x-1)$). Thus, the average node multiplicity in $D(\mathcal{H})[X']$ is $\frac{2e_{D(\mathcal{H})}(X')}{|X'|} \geq (2x-2) + \frac{2(y-1)(x+1)}{Cx+y-1} \geq 2x - 2$. Consequently, either there is a node of multiplicity at least $2x - 1$ in $D(\mathcal{H})[X']$ or all nodes X' have multiplicity $2x - 2$. In $D(\mathcal{H})[X]$, at least one node from X' has an additional edge to v. Hence, there must be $u \in X'$ with $m_{D(\mathcal{H})}(\{u\}, X) \geq 2x - 1 = \sigma^C_{Cx+y} - 1$.

Since $X \setminus \{u\}$ is not a Strong Attack, it holds that $e_{D(\mathcal{H})}(X \setminus \{u\}) \geq \sum_{i=1}^{Cx+y-1} \sigma_i^C$. However, X is strong and induces $m_{D(\mathcal{H})}(\{u\}, X)$ additional edges. We obtain

$$\sigma^C_{Cx+y} - 1 + \sum_{i=1}^{Cx+y-1} \sigma_i^C \leq m_{D(\mathcal{H})}(\{u\}, X) + e_{D(\mathcal{H})}(X \setminus \{u\}) = e_{D(\mathcal{H})}(X) < \sum_{i=1}^{Cx+y} \sigma_i^C. \tag{4.71}$$

Inequality 4.71 confirms that $e_{D(\mathcal{H})}(X) = Cx(x-1)+2xy-1$, thereby proving Property 3. Furthermore, it enforces that $m_{D(\mathcal{H})}(\{u\}, X) = 2x - 1$. The same argument applies to all other $v \in X$ with $m_{D(\mathcal{H})}(\{v\}, X) \geq 2x - 1$. We obtain Property 4.

Due to the multiplicity upper bound of Property 4, the attack X can induce at most $e_{D(\mathcal{H})}(X) \leq (Cx + y)(2x - 1)/2$ edges. However, for $y > \frac{Cx+2}{2x+1}$, this is less than the $Cx(x-1) + 2xy - 1$ edges necessary for Property 3. We obtain Property 2.

Finally, we prove the upper bound in Property 1. For this, we show that if X is a Minimum Strong Attack, then $\bar{X} := V \setminus X$ is also a Strong Attack. Due to Property 3, the fact that $D(\mathcal{H})$ is $2(k-1)$-partite, and $|A| = Ck(k-1)$, it holds that

$$\begin{aligned}
e_{D(\mathcal{H})}(\bar{X}) &= |A| - |X| \cdot 2(k-1) + e_{D(\mathcal{H})}(X) \\
&= Ck(k-1) - (Cx+y)2(k-1) + Cx(x-1) + 2xy - 1 \\
&= (k-1)(Ck - 2Cx - 2y) + Cx(x+1) - 2x(C-y) - 1 \\
&= (k-1)(Ck - 2Cx - 2C) + Cx(x+1) + (C-y)(2(k-1) - 2x) - 1 \\
&= C\left((k-1)(k-2x-2) + x(x+1)\right) + 2(k-x-1)(C-y) - 1 \\
&= C\left((k-x-1)(k-x-2)\right) + 2(k-x-1)(C-y) - 1 \\
&= \left(\sum_{i=1}^{C(k-x-1)+(C-y)} \sigma_i^C\right) - 1 = \left(\sum_{i=1}^{|\bar{X}|} \sigma_i^C\right) - 1.
\end{aligned} \tag{4.72}$$

If $|X| > Ck/2$, it holds that $\bar{X} \leq Ck/2 < |X|$. Hence, $\bar{X}$ would be a Strong Attack smaller than X. This is impossible. □

Using Lemma 4.3.15, we can identify additional optimally LiSS-stable head topologies.

Corollary 4.3.16
Let $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ be a head topology with subtopologies $\mathcal{H}_1, \ldots, \mathcal{H}_r$, the properties given in Corollary 4.3.2, and that meets the Stability Requirements. For each subtopology $\mathcal{H}_i$, there is C_i with $\mathcal{H}_i \in \mathbb{T}(C_i k, C_i, k)$.

Head topology $\mathcal{H}$ is optimally LiSS-stable, if $k \leq 4$ and $\forall i \in [r]\colon C_i \leq 3$.

Proof. Each Minimum Strong Attack X on an arbitrary subtopology $\mathcal{H}_i$ has the properties listed in Lemma 4.3.15. With $k \leq 4$, Property 1 demands that $x = 1$ and $y \geq 2$. However, since $C_i \leq 3$, the latter conflicts with Property 2. The absence of Minimum Strong Attacks makes all subtopologies optimally LiSS-stable. The stability of $\mathcal{H}$ follows from Lemma 4.3.7. □

The Figures 4.10(a) and 4.10(b) show dependency graphs of such topologies.

A restriction to head topologies with $k \leq 4$ has yet another advantage. We show that it leads to the fact that Minimum Strong Attacks will induce exactly 3 edges in the topology's dependency graph.

Lemma 4.3.17
Let $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ be a head topology with $k \leq 4$, conforming to the Stability Requirements and with the properties given in Corollary 4.3.2.

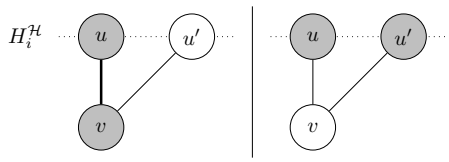
If $\mathcal{H}$ is not optimally LiSS-stable, then there is a Minimum Strong Attack $X^3 \subseteq V$ of cardinality $|X^3| = C + 2$, such that $D(\mathcal{H})[X^3]$ has exactly three edges $\{u_1, v_1\}, \{u_2, v_2\}$ and $\{w_1, w_2\}$. Additionally, in $D(\mathcal{H})$, each node in these edges can be reached from w_1 by a path over at most 3 hops.

Proof. Let X be an arbitrary Minimum Strong Attack on $\mathcal{H}$. There are $x \in [0, k-1]$ and $y \in [C]$ such that $|X| = Cx + y$. Due to Lemma 4.3.15, we know that $x = 1$ and that $D(\mathcal{H})[X]$ has $2y - 1$ edges. Additionally, these edges are a matching in $D(\mathcal{H})$. We split X into the set X_e containing the $2(2y-1)$ nodes incident to induced edges and the set $X_{\bar{e}} := X \setminus X_e$.

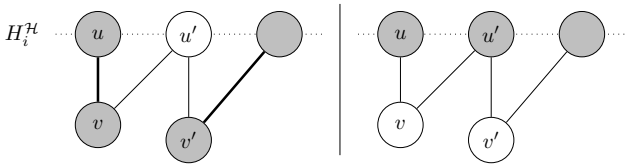
Since $D(\mathcal{H})$ is k-partite with the head sets $H_1^{\mathcal{H}}, \ldots, H_k^{\mathcal{H}}$ as parts, each induced edge contains heads from two different stripes. On average, each head set is incident to at least $2(2y-1)/k \geq 1/2 \cdot (2y-1) = y - 1/2$ of these egdes. Thus, there is a stripe $i \in [k]$ such that $H_i^{\mathcal{H}}$ is incident to at least y, i.e., more than half, of the induced edges.

Now, let $\{u, v\}$ be an edge of $D(\mathcal{H})[X]$ with $u \in H_i^{\mathcal{H}}$. Since $\mathcal{H}$ has the properties given in Corollary 4.3.2, v has *one* other edge to a node $u' \in H_i^{\mathcal{H}} \setminus X$. Distinguish the following cases:

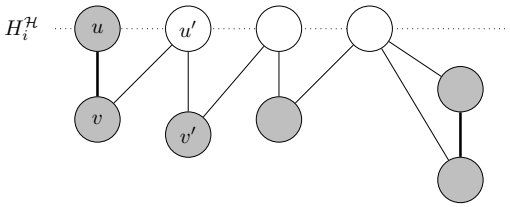
1. $|N(u') \cap X| = 1$: This case cannot appear. Otherwise, the set $X' := X \setminus \{v\} \cup \{u'\}$ is a Strong Attack of cardinality $|X|$ that induces one edge less than X. Thus, X would not be a Minimum Strong Attack. The difference between both attacks is sketched in Figure 4.12(a).



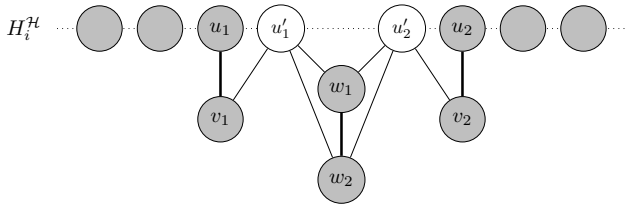
(a) Difference of attacks X and X' in Case 1.



(b) Difference of attacks X and X' in Case 2a.



(c) The Case 2b.



(d) Scheme of attack $X^3 = X_e^3 \cup X_{\bar{e}}^3$.

Figure 4.12.: Illustrations for the proof of Lemma 4.3.17 (nodes from X resp. X' gray, only edges between depicted nodes shown).

2. $|N(u') \cap X| = 2$: Again, distinguish the following cases:
   a) $N(u') \subseteq X_e$: This case cannot appear. Here, u' is neighbored to a second node v' that is incident to an edge in $D(\mathcal{H})[X]$. Then, the set $X' := X \setminus \{v, v'\} \cup \{u'\}$ is an attack of cardinality $|X| - 1$ that induces $2 = \sigma^C_{|X|}$ edges less than X. Thus, X was not a *Minimum* Strong Attack, since X' is a Strong Attack, too. The difference between both attacks is sketched in Figure 4.12(b).
   b) $N(u') \cap X = \{v, v'\}$ with $v' \in X_{\bar{e}}$: This case may appear. An example is shown in Figure 4.12(c). We must have $m_{D(\mathcal{H})}(u', v') = 1$, because if $\{u', v'\}$ was a parallel edge, the Stability Requirements would enforce the existence of an edge $\{v, v'\}$. Since $v' \in X_{\bar{e}}$, such an edge does not exist.

      Note, that we can '*move the induced edge to* u'' by modifying X to be $X' := X \setminus \{v\} \cup \{u'\}$. Then, X' is also a Minimum Strong Attack. Furthermore, if Case 2b applies to the 'new' edge $\{u', v'\}$, too, we can continue this process. Since, each time, it holds that $|N(u') \cap X| = 2$ and $|N(v) \cap H_i^{\mathcal{H}}| = 2$, the nodes u', v from the iterated process form a (loopless) path that is unique for the original edge $\{u, v\}$. Since $|H_i^{\mathcal{H}} \cap X_{\bar{e}}|$ is finite, we eventually obtain a set X' in which one of the remaining cases applies to u'. Furthermore, since the nodes on the path of $\{u, v\}$ cannot be contained in a path for another edge incident to $H_i^{\mathcal{H}}$, we can move *all* such edges and obtain an X' in which Case 2b does not appear anymore.

3. $|N(u') \cap X| = 3$: This case may appear. Let $\{v, w_1, w_2\} = N(u') \cap X$. Due to the claw-freeness of $D(\mathcal{H})$, w_1 and w_2 induce an edge $\{w_1, w_2\}$ in $D(\mathcal{H})$. Since $u' \in H_i^{\mathcal{H}}$, it holds that $w_1, w_2 \notin H_i^{\mathcal{H}}$.

4. $|N(u') \cap X| \geq 4$: This case cannot appear. The node v already induces an edge together with u. Therefore, it has no edges with the nodes $N(u') \cap X$. To achieve the claw-freeness of $D(\mathcal{H})$, the nodes $X \cap N(u') \setminus N(v)$ needed to induce a clique. However, the maximum multiplicity of a node in $D(\mathcal{H})[X]$ is one.

We see that X can be transformed, such that only Case 3 occurs. Assume that this has been done.

In $D(\mathcal{H})[X]$, we have at least y edges incident to $H_i^{\mathcal{H}}$ and between 1 and $y - 1$ edges not incident to a node from $H_i^{\mathcal{H}}$. Since each of the former has a neighbor $u' \in H_i^{\mathcal{H}}$ as in Case 3 above, there exists an edge $\{w_1, w_2\}$ of the latter set, such that there are *two* distinct induced edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$ with $u_1, u_2 \in H_i^{\mathcal{H}}$ and nodes $u'_1, u'_2 \in H_i^{\mathcal{H}} \setminus X$ with $\{w_1, w_2\} \subseteq N(u'_1) \cap N(u'_2)$. Since w_1 and w_2 both have multiplicity 2 to $H_i^{\mathcal{H}}$, u'_1 and u'_2 are the only neighbors of $\{w_1, w_2\}$ in $H_i^{\mathcal{H}}$. See Figure 4.12(d) for an example. For $X^3_e := \{u_1, v_1, u_2, v_2, w_1, w_2\}$, we then have $|(X^3_e \cup N(X^3_e)) \cap H_i^{\mathcal{H}}| = 4$ and the nodes $X^3_{\bar{e}} := H_i^{\mathcal{H}} \setminus (X^3_e \cup N(X^3_e))$ are an independent set of $D(\mathcal{H})$ of cardinality $C - 4$. Consequently, the set $X^3 = X^3_e \cup X^3_{\bar{e}}$ has cardinality $C + 2$ and induces 3 edges in $D(\mathcal{H})$. Given that no Minimum Strong Attack of cardinality $C + 1$ exists (see Lemma 4.3.15), it is a Minimum Strong Attack. □

Lemma 4.3.17 shows that for small k, the number of edges induced by a Minimum Strong Attack is actually independent of the parameter C. It seems possible, that this value is generally only depending on k. However, a corresponding proof has not yet been found. For the moment, we can now efficiently determine the LiSS-stability of head topologies with $k \leq 4$.

Corollary 4.3.18
The LiSS-Stability Decision Problem for head topologies $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ with $k \leq 4$ can be solved in time in $O(C)$.

Proof. First, we check for the properties listed in Corollary 4.3.2. With a tree traversal, this is possible time $O(Ck^2)$. Since $k = O(1)$, we have $O(Ck^2) = O(C)$.

Then, we verify adherence to the Stability Requirements and use Lemma 4.3.17 to determine whether a Mininmum Strong Attack exists. Both checks can be made by evaluating certain constant-size neighborhoods for each $v \in V$ in $D(\mathcal{H})$. In particular, let $N(v, \leq 3)$ be the nodes with a path of length at most 3 to v in $D(\mathcal{H})$. With $k = O(1)$, we have $\forall v \in V\colon m_{D(\mathcal{H})}(v) = 2(k-1) = O(1)$ and $\forall v \in V\colon |N(v, \leq 3)| = O(1)$. For each $v \in V$, we check $D(\mathcal{H})[N(v, \leq 3)]$ for claw-freeness, the necessary multiplicities between the nodes $N(v)$, and whether there is a combination of 3 edges as induced by the set X_e^3 in the proof of Lemma 4.3.17. Since $|V| = Ck = O(C)$, the whole process needs $O(C)$ time. □

With these results, the remaining head topologies for which we cannot yet efficiently decide their LiSS-stability are head topologies with $k \geq 5$, that have the properties given in Corollary 4.3.2, satisfy the Stability Requirements, but are not line graphs.

To solve the LiSS-Stability Decision Problem on these topologies, we currently have to retreat to an exhaustive search for Strong Attacks. However, the search space can be restricted to *Minimum* Strong Attacks. This allows to use the results of Lemma 4.3.15 and Corollary 4.3.14 to heavily confine the parameters of such a search (e.g., attack cardinality, multiplicity of nodes in the induced subgraph, number of attacked stripes).

4.4. Complexity of the LiSS-Stability Decision Problem

In the preceding sections, we have obtained both a characterization of the optimally LiSS-stable topologies in $\mathbb{T}(n, C, k)$ and quite a number of efficiently checkable, necessary or sufficient conditions on these topologies. Now, we investigate whether the general LiSS-Stabiliy Decision Problem can be solved efficiently, too. We show that this would imply $\mathbf{P} = \mathbf{NP}$.

To prove this results, we first introduce the decision version of another **NP**-complete problem ([GJ79] problem '[GT20]').

Definition 4.4.1 ***Independent Set Problem***
Given a graph $G = (V, E)$ and a number $t \in [\,|V|\,]$, decide whether there exists $X \subseteq V$ such that $|X| = t$ and $\forall u, v \in X\colon \{u, v\} \notin E$.

The Independent Set Problem is trivial for $t = 1$: since each set $\{v\}$ with $v \in V$ does not induce an edge in G, the solution is *yes* if and only if $|V| \neq \emptyset$. Hence, the problem clearly remains **NP**-complete when restricted to instances with $t > 1$.

The following result was first shown by Andreas Brieg in [Bri08] and is published in [BBG+09].

Theorem 4.4.2
The LiSS-Stability Decision Problem is **coNP**-complete.

Proof. To prove **coNP**-completeness of the LiSS-Stability Decision Problem, it is sufficient to prove **NP**-completeness for its inverse problem:

Definition 4.4.3 LiSS-*Instability Decision Problem*
Given a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, decide whether there is a Strong Attack $X \subseteq V$ on $\mathcal{T}$ or not.

Due to Theorem 4.1.9, $\mathcal{T}$ is optimally LiSS-stable if and only if there is *no* Strong Attack on $\mathcal{T}$. Hence, both problems are indeed inverse.

The LiSS-Instability Decision Problem is in **NP**, since we can validate a guessed attack $X \subseteq V$ by checking whether $a^{\mathcal{T}}(X) > \sum_{i=1}^{|X|} \delta_i^{C,k}$. The latter can be computed in time $O(kn)$ by traversing all trees. To show **NP**-completeness, we give a polynomial-time reduction from the Independent Set Problem.

Let $(G = (K, A), t)$ be an instance of the Independent Set Problem with $t > 1$ and let $n_G := |K| > 1$. W.l.o.g. assume $K = [n_G]$. From (G, t), we construct a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ with

$$n := (t-1)(2(n_G - 1)(4n_G(n_G - 1) - 1) + 1), \tag{4.73}$$
$$C := t - 1, \tag{4.74}$$
$$k := 2n_G(n_G - 1). \tag{4.75}$$

In particular, for each tupel (u, v) with $u \in V, v \in V \setminus \{u\}$, we can define a tree index function f as

$$\mathrm{f}(u, v) := \begin{cases} 2((u-1)(n_G - 1) + v) & , \text{ if } v < u \\ 2((u-1)(n_G - 1) + v - 1) & , \text{ otherwise.} \end{cases} \tag{4.76}$$

Its definition guarantees, that for each such tupel (u, v) there are two *unique* stripes $T_{\mathrm{f}(u,v)}$ and $T_{\mathrm{f}(u,v)+1}$ in $\mathcal{T}$. The node set $V = [n]$ of $\mathcal{T}$ is partitioned into the sets

$$K = [n_G] \tag{4.77}$$
$$H^{\mathcal{T}} = [n_G + 1, n_G + Ck] \tag{4.78}$$
$$D_e^1 = [\, n_G + Ck + 1, n_G + Ck + k \,] \tag{4.79}$$
$$D_e^2 = [\, n_G + Ck + k + 1, n_G + Ck + k + \left(\frac{n}{C} - 2k - 2\right)] \tag{4.80}$$
$$D_o = V \setminus K \setminus H^{\mathcal{T}} \setminus D_e^1 \setminus D_e^2. \tag{4.81}$$

Since $n_G \geq 2$ and $t \geq 2$, none of these sets is empty. Note, that the nodes K will *not* be used as heads. As a general rule, the nodes $D_e^1 \cup D_e^2 \cup D_o$ will forward under no circumstances and a head $h \in H_i^{\mathcal{T}}$ is childless in all stripes T_j with $j \neq i$.

The head topology $\mathcal{H}$ of $\mathcal{T}$ is set to be a Cluster Topology in $\mathbb{T}(Ck, C, k)$ (see Definition 4.1.6). Consequently, there are C clusters of heads $H^{\mathcal{T}(1)}, \ldots, H^{\mathcal{T}(C)}$, each containing exactly one head of each stripe, i.e., for $i \in [C], j \in [k]$ the set $H_j^{\mathcal{T}(i)} := H^{\mathcal{T}(i)} \cap H_j^{\mathcal{T}}$ has cardinality one.

In the construction of $\mathcal{T}$, we ensure that for each head cluster $r \in C$ and each node $u \in K$, there is at least one $v \in K \setminus \{u\}$, such that in $T_{\mathrm{f}(u,v)}$ and $T_{\mathrm{f}(u,v)+1}$, the node u is a *child* of a head from $H^{\mathcal{T}(r)}$. This is always possible, since $|K \setminus \{u\}| = n_G - 1$ and $C = t - 1$ with $t \leq n_G$.

For each tupel (u, v) with $u \in K$ and $v \in K \setminus \{u\}$, the layout of $T_{\mathrm{f}(u,v)}$ and $T_{\mathrm{f}(u,v)+1}$ depends on whether $\{u, v\} \in A$ or not. Figure 4.13 visualizes the trees built for both cases. For $i \in \{0, 1\}$, let h_i be the first (and only) head on the $s \to u$ path in $T_{\mathrm{f}(u,v)+i}$ and let $H^{\mathcal{H}(q)}$ be its cluster.

If $\{u, v\} \in A$, then we set

$$\mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(h_i) := \{u\} \cup D_e^1 \cup D_e^2, \tag{4.82}$$

$$\mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(u) := \{v\} \text{ and} \tag{4.83}$$

$$\mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(v) := H^{\mathcal{H}(q)} \setminus \{h_i\}. \tag{4.84}$$

Alternatively, with a node $d \in D_e^1$, we set

$$\mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(h_i) := \{u, v\} \cup D_e^2, \tag{4.85}$$

$$\mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(u) := H^{\mathcal{H}(q)} \setminus \{h_i\} \cup \{d\} \text{ and} \tag{4.86}$$

$$\mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(v) := D_e^1 \setminus \{d\}. \tag{4.87}$$

In both cases, it holds that

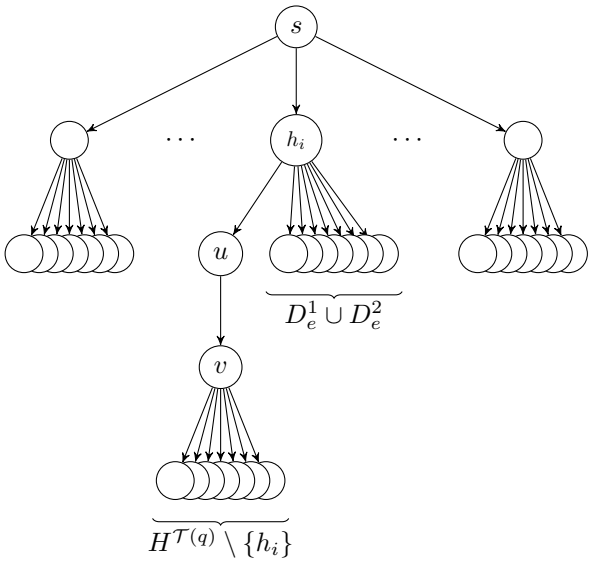
$$|\mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(h_i)| = |D_e^2| + |D_e^1| + |H^{\mathcal{H}(q)}| + |\{u, v\}| = \frac{n}{C} - 2k - 2 + k + k + 2 = \frac{n}{C}, \tag{4.88}$$

as well as

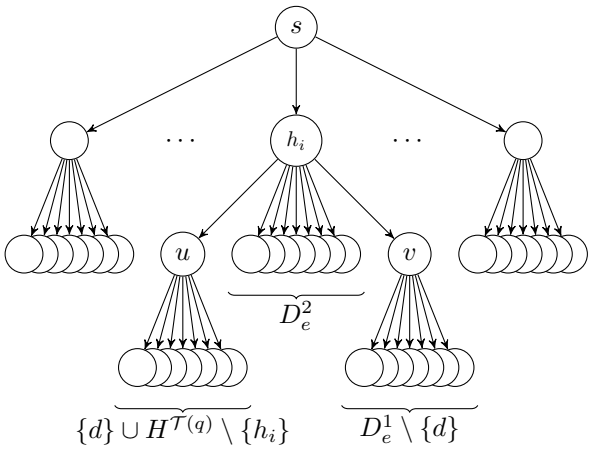
$$|\mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(u)| = k + 1 \quad \text{and} \quad |\mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(v)| = k. \tag{4.89}$$

The $(C-1)\left(\frac{n}{C} - 1\right)$ nodes $V \setminus H^{\mathcal{T}}_{\mathrm{f}(u,v)+i} \setminus H^{\mathcal{T}(q)} \setminus \{u, v\} \setminus D_e^1 \setminus D_e^2$ are positioned as children to the $C - 1$ remaining heads $H^{\mathcal{T}}_{\mathrm{f}(u,v)+i} \setminus \{h_i\}$ of stripe $T_{\mathrm{f}(u,v)+i}$. In particular, it is ensured that $\forall h_j \in H^{\mathcal{T}}_{\mathrm{f}(u,v)+i}\colon |\mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(h_j)| = \frac{n}{C}$ holds and that the clustered head topology $\mathcal{H}$ is obtained.

The resulting topology $\mathcal{T}$ has $O(n_G^4)$ peers and $O(n_G^6)$ edges. The transformation from G to $\mathcal{T}$ consists of a successor assignment satisfying the aforementioned constraints and is computable in time $O(n_G^6)$.



(a) Scheme of $T_{\mathrm{f}(u,v)+i}$ if $\{u,v\} \in A$ $\left(\text{with} \{h_i\} = H^{\mathcal{T}(q)}_{\mathrm{f}(u,v)+i}\right)$.



(b) Scheme of $T_{\mathrm{f}(u,v)+i}$ if $\{u,v\} \notin A$ $\left(\text{with} \{h_i\} = H^{\mathcal{T}(q)}_{\mathrm{f}(u,v)+i}\right)$.

Figure 4.13.: A tree $T_{\mathrm{f}(u,v)+i}$ in the proof of Theorem 4.4.2 ($i \in \{0,1\}$).

The node set partition influences the value $a^{\mathcal{T}}(v) = \sum_{i=1}^{k} |\mathrm{succ}_i^{\mathcal{T}}(v)|$ for each $v \in V$:

- $v \in D_e^1 \cup D_e^2 \cup D_o$: Node v appears only as leaf in $\mathcal{T}$. By assumption, we have $1 < t \leq n_G$ and $n_G \geq 2$. Hence, we obtain

$$a^{\mathcal{T}}(v) = k < 2(n_G - 1)(4n_G(n_G - 1) - 1) + 1 - 2((n_G - 1) - 1) + k - 1 \quad (4.90)$$
$$= \frac{n}{C} - 2(C - 1) + k - 1 = \delta_{Ck}^{C,k}. \quad (4.91)$$

- $v \in K$: Node v has $k + 1$ successors in each stripe $T_{\mathrm{f}(v,u)+i}$ for $i \in \{0, 1\}$ and $u \in K \setminus \{v\}$, another k successors in each stripe $T_{\mathrm{f}(u,v)+i}$, and no successors besides itself in the remaining stripes. Since $k = 2n_G(n_G - 1)$, this adds up to

$$a^{\mathcal{T}}(v) = 2(n_G - 1)(k + 1) + 2(n_G - 1)k + k - 4(n_G - 1) \quad (4.92)$$
$$= 2(n_G - 1)(4n_G(n_G - 1) - 1) + k \quad (4.93)$$
$$= 2(n_G - 1)(4n_G(n_G - 1) - 1) + 1 + k - 1 \quad (4.94)$$
$$= \frac{n}{C} + k - 1 = \delta_1^{C,k}. \quad (4.95)$$

- $v \in H^{\mathcal{T}}$: There is exactly one $i \in [k]$ with $v \in H_i^{\mathcal{T}}$. Head v has $|\mathrm{succ}_i^{\mathcal{T}}(v)| = \frac{n}{C}$ and $\forall j \in [k] \setminus \{i\}\colon |\mathrm{succ}_j^{\mathcal{T}}(v)| = |\{v\}| = 1$. Hence, it holds that $a^{\mathcal{T}}(v) = \delta_1^{C,k}$.

Topology $\mathcal{T}$ has the following property.

Claim 4.4.4
There is a Strong Attack $X \subseteq V$ on $\mathcal{T}$ if and only if there is an independent set of cardinality t in G.

Proof. "If": Assume $G = (K, A)$ has an independent set $I \subseteq K$ with $|I| = t = C + 1$. Due to the construction of $\mathcal{T}$, for each two nodes $u, v \in K$ with $\{u, v\} \notin A$, it holds that $\forall i \in [k]\colon \mathrm{succ}_i^{\mathcal{T}}(u) \cap \mathrm{succ}_i^{\mathcal{T}}(v) = \emptyset$. Since the sequence $(\delta_i^{C,k})_{1 \leq i \leq Ck}$ is non-increasing and $\delta_{C+1}^{C,k} = \delta_1^{C,k} - 2$, we obtain

$$a^{\mathcal{T}}(I) = \sum_{i=1}^{k} \sum_{v \in I} |\mathrm{succ}_i^{\mathcal{T}}(v)| = \sum_{v \in I} a^{\mathcal{T}}(v) = (C + 1)\delta_1^{C,k} > \sum_{i=1}^{C+1} \delta_i^{C,k}. \quad (4.96)$$

Consequently, I is a Strong Attack on $\mathcal{T}$.

"Only-If": Assume G does *not* have an independent set of cardinality t. To rule out the existence of any Strong Attack on $\mathcal{T}$, it is sufficient to show the inexistence of a *Minimum* Strong Attack. Due to Lemma 4.1.11 and Inequality 4.91, such a Minimum Strong Attack X must have $X \subseteq H^{\mathcal{T}} \cup K$. Distinguish the following cases:

1. $X \subseteq H^{\mathcal{T}}$: Topology $\mathcal{T}$ adheres to the rules Not-Too-Many-Successors, Head Rule 1, Head Rule 2 and Heads-Are-Optimally-Stable from Section 4.2. Hence, Lemma 4.2.2 rules out the existence of such Strong Attacks.

2. $X \subseteq K$: It holds that $\forall v \in V : a^{\mathcal{T}}(v) \leq \delta_1^{C,k} = \delta_C^{C,k}$. Hence, there is no Strong Attack X with $|X| \leq C = t - 1$, since it follows that

$$\forall X \subseteq V, |X| \leq C : a^{\mathcal{T}}(X) \leq \sum_{v \in X} a^{\mathcal{T}}(v) \leq \sum_{i=1}^{|X|} \delta_i^{C,k}. \tag{4.97}$$

Furthermore, since G does not have an independent set of cardinality t, every set $X \subseteq K$ with $|X| > C$ contains nodes $u, v \in K$ with $\{u, v\} \in A$. For $i \in \{0, 1\}$, we then have $|\mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(u) \cap \mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(v)| = k = |\mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(v,u)+i}(u) \cap \mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(v,u)+i}(v)|$.

Since $4k = 8n_G(n_G - 1) > 2(n_G - 1) > 2(t - 1) - 2 = 2(C - 1)$, this leads to

$$a^{\mathcal{T}}(X) - a^{\mathcal{T}}(X \setminus \{u\}) \leq a^{\mathcal{T}}(u) - 4k = \delta_1^{C,k} - 4k < \delta_{Ck}^{C,k}. \tag{4.98}$$

However, then Lemma 4.1.11 implies that X is not a Minimum Strong Attack.

3. $X \cap H^{\mathcal{T}} \neq \emptyset \wedge X \cap K \neq \emptyset$: As seen in the last case, if X is a Minimum Strong Attack, it must have $|X \cap K| \leq C$.

For each head cluster $H^{\mathcal{T}(q)}$ with $q \in [C]$, each $u \in X \cap K$, and each $i \in \{0, 1\}$, the construction of $\mathcal{T}$ guarantees that there is at least one node $v \in K \setminus \{u\}$ such that there is a head $h_i \in H^{\mathcal{T}(q)}$ with $u \in \mathrm{child}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(h_i)$. Due to the head clustering, it also holds that $H^{\mathcal{T}(q)} \setminus \{h_i\} \subseteq \mathrm{succ}^{\mathcal{T}}_{\mathrm{f}(u,v)+i}(u)$. Additionally, there is no stripe in $\mathcal{T}$ in which the successor set of u contains elements from the successor sets of more than one head cluster. Consequently, for $u \in X \cap K$, we can write

$$\sum_{i=1}^{k} |\mathrm{succ}_i^{\mathcal{T}}(u) \cap \mathrm{succ}_i^{\mathcal{T}}(X \cap H^{\mathcal{T}})| = \sum_{i=1}^{k} \sum_{q \in [C]} |\mathrm{succ}_i^{\mathcal{T}}(u) \cap \mathrm{succ}_i^{\mathcal{T}}(X \cap H^{\mathcal{T}(q)})| \tag{4.99}$$

$$\geq \sum_{q \in [C]} 2|X \cap H^{\mathcal{T}(q)}| \tag{4.100}$$

$$\geq 2|X \cap H^{\mathcal{T}}| = 2|X \setminus K|. \tag{4.101}$$

This leads to

$$a^{\mathcal{T}}(X) - a^{\mathcal{T}}(X \setminus \{u\}) \leq a^{\mathcal{T}}(u) - 2|X \setminus K| \tag{4.102}$$

$$\leq \delta_1^{C,k} - 2\max(1, |X| - C + 1) \tag{4.103}$$

$$< \delta_1^{C,k} - 2\left\lfloor \frac{|X| - 1}{C} \right\rfloor = \delta_{|X|}^{C,k}. \tag{4.104}$$

Again, Lemma 4.1.11 shows that X is not a Minimum Strong Attack.

Since none of the three possible cases permits the existence of a Minimum Strong Attack, no Strong Attack can exist on $\mathcal{T}$. □

Claim 4.4.4 confirms that we have shown a polynomial-time reduction from Maximum Independent Set to the LiSS-Instability Decision Problem. Since the latter is in **NP**, it must be **NP**-complete. Consequently, the inverse LiSS-Stability Decision Problem is **coNP**-complete. □

We see, that the existence of a polynomial-time algorithm for the LiSS-Stability Decision Problem would induce that $\mathbf{P} = \mathbf{coNP}$ and $\mathbf{P} = \mathbf{NP}$. Currently, this is considered to be highly unlikely.

However, the topology management of a peer-to-peer live streaming system must be able to efficiently decide about the LiSS-stability of its maintained topology. Thus, it is forced to build topologies for which this is possible. With the Cluster Topologies of Section 4.1.3 and the Rule-Based Topologies of Section 4.2, we identified two large subsets of the optimally LiSS-stable topologies for which membership can be checked in polynomial time. The distributed implementation of these topologies will be the topic of Section 4.5

A question still unaddressed by the proof of Theorem 4.4.2, is the complexity of the LiSS-Stability Decision Problem for head topologies. Here, in Section 4.3, we were able to identify a large number of head topologies for which this problem is solvable in polynomial-time. However, the problem's complexity remains unknown on head topologies $\mathcal{H}$ with $k \geq 5$ that have the properties listed in Corollary 4.3.2, adhere to the Stability Requirements, but are not line graphs.

We have seen, that the LiSS-Stability Decision Problem for such a head topology $\mathcal{H} \in \mathbb{T}(Ck, C, k)$ corresponds to deciding whether there is no induced subgraph of $D(\mathcal{H})$ that has $Cx + y$ ($x \in [0, k-1], y \in [C]$) nodes but less than $Cx(x-1) + 2xy$ edges. Finding such a subgraph was shown to be **NP**-complete on general graphs, cubic and planar graphs [Yan78]. Furthermore, the similar problem asking for the Maximum Induced Matching in a claw-free graph is also **NP**-complete [KR03].

However, the studied multigraphs are *highly structured*, so that a general solvability in polynomial-time does not seem implausible. This impression is supported by the progress we already made in Section 4.3. A promising starting point for further results seems to be the search for a generalization of Lemma 4.3.17.

4.5. Heuristics for a Distributed Implemention of Optimally LiSS-Stable Topologies

A topic we have yet given only little attention, is how optimally LiSS-stable topologies can be built in peer-to-peer systems without global coordination. This is quite an important aspect, since the presence of a central topology-coordinating entity with its inherently limited computing and communication resources would also limit the maximum number of peers in the system. However, overcoming such limitations is one of the main motivations to introduce peer-to-peer based streaming system.

Consequently, we now sketch two possible approaches to implement optimally LiSS-stable topologies using distributed topology management mechanisms. For reasons of space and topical focus, we will not give an exhaustive description but restrict to the

presentation of key ideas. Details of a (manipulation-resistant) implementation are given in [BFGS09a, BSS09, Fis12]. Note that these approaches are merely heuristics and *cannot guarantee* the formation of optimally LiSS-stable topologies. In particular, depending on the bandwidth resources of the participating peers, there are scenarios in which this is generally impossible.

Tree Balancing A basic building block of both heuristics is a mechanism to balance individual stripe trees. To gather the necessary information, each node $v \in V$ in each stripe $i \in [k]$ has to continuously report the value $|\text{succ}_i^{\mathcal{T}}(v)|$ to its parent $\text{parent}_i^{\mathcal{T}}(v)$. The latter accumulates these numbers to calculate its own successor number. Based on this information, each peer can calculate the average successor number among its children and initiate node movements between the successor sets of children deviating from it.

Cluster Topologies Introducing two additional mechanisms, we can now aim at forming Cluster Topologies. For this, the source and the bootstrapping server act as if they would manage C *peer-disjoint* peer-to-peer live streaming systems distributing the same stream. The node sets of these systems are $V_1, \ldots, V_C$. Based on successor number information obtained from the source, the bootstrapping server then distributes joining nodes to the system with the smallest node set. Assuming an equal node departure behaviour in each of the sets $V_1, \ldots, V_C$, this leads to approximately balanced cardinalities.

Furthermore, based on its current number of successors in each stripe, each node chooses a *favorite* stripe. Then, it constantly strives to hand over successors in the remaining stripes to other nodes. Measurements in [BSS09] confirm, that this technique leaves only a small number of nodes $(4 - 10\%)$ forwarding in more than one stripe. Furthermore, these nodes are restricted to a very small number of successors.

Applying both techniques together leads to topology properties that are close to the definition of Cluster Topologies.

However, Cluster Topologies enforce heavy restrictions on the possible successor relationships in the peer-to-peer streaming system. In that, they make it unnecessarily difficult to properly exploit the bandwidth resources of all participating peers. Additionally, we will see in Chapter 5, that they are inherently unstable regarding the LoSS-damage measure. Hence, it is more desirable to aim at constructing the less restrictive rule-based topologies from Section 4.2.

Rule-Based Topologies Once again, a small set of mechanisms suffices to approximate the respective rules. The most difficult problem is posed by implementing the Strictly-Not-Too-Many-Successors rule. Of course, this could be done by publicly announcing actual successor number limits towards the leafs. However, such an approach reveals important topology information and allows nodes to estimate their influence on the whole system.

A possible indirect solution, adopts the technique of 'favorite stripes' we already applied for Cluster Topologies. The measurements in [BSS09] show, that the important nodes near the source will then forward in only one stripe. In combination with the aforementioned stripe tree balancing, this leads to a balanced division of the successor number of each parent (reduced by one) among its children. Assuming $n \gg Ck$ and that heads have sufficient bandwidth resources to support at least 2 children, it is possible to approximate the required successor limit for heads and effectively realize them for non-heads.

The implementation of the Head Rule 1, Head Rule 2 and the Heads-Are-Optimally-Stable rule is simplified by the facts, that all heads are, in some stripe, children of the source and that their number is limited to Ck. Therefore, it is possible to burden these rules' coordination on the source without creating scalability issues. Here, the necessary information about the head topology can be obtained by sending, in each stripe i and for each head $h \in H_i^{\mathcal{T}}$, a unique label along with the stripe data. The received labels are collected from the heads and document their supply relationships [GFS11]. Since successor number information is already available via the tree balancing mechanism, the source can then initiate the necessary rearrangements of the heads' dependencies to satisfy the head-related rules.

4.6. Summary

In this chapter, we studied distribution topologies minimizing, for each possible cardinality of attacks, the maximum LiSS-damage that can be dealt to them.

In Subsection 4.1.1, these topologies were named optimally LiSS-stable topologies. After motivating our approach, we introduced formal specifications of the studied problems. The Optimally LiSS-stable Topology Formation Problem consists in finding an optimally LiSS-stable topology from a given class $\mathbb{T}(n, C, k)$. In the LiSS-Stability Decision Problem, we want to determine whether a given topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is optimally LiSS-stable. In the course of this chapter, we saw that both are of very different computational complexity if $\mathbf{P} \neq \mathbf{NP}$.

In the following Subsections 4.1.2 and 4.1.3, we reviewed results of [BSS09]. First, we introduced the damage sequence $(\delta_i^{C,k})_{1 \leq i \leq Ck}$. Then, we showed that there is a simple greedy polynomial-time algorithm that, when run on a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ and a number $x \in [n]$, returns an attack X with $|X| = x$ and $a^{\mathcal{T}}(X) \geq \sum_{i=1}^{x} \delta_i^{C,k}$.

This observation was complemented by the identification of the Cluster Topologies. Every topology $\mathcal{T}$ from this non-empty subclass of $\mathbb{T}(n, C, k)$ has the property $\forall X \subseteq V \colon a^{\mathcal{T}}(X) \leq \sum_{i=1}^{|X|} \delta_i^{C,k}$. With this result, we both obtained a damage-based characterization of optimally LiSS-stable topologies and found a first class of topologies that indeed are optimally LiSS-stable. Membership in the class of Cluster Topologies is checkable in polynomial time.

Based on the obtained characterization, we then identified necessary properties of optimally LiSS-stable topologies in Subsection 4.1.4. They were used in Section 4.2 to state a set of polynomial-time checkable rules that define a new, less restrictive subclass of optimally LiSS-stable topologies.

Given a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, one of these rules is concerned with the head topology $\mathcal{H}$ of $\mathcal{T}$. The head topology is obtained from $\mathcal{T}$ by removing the nodes $V \setminus H^{\mathcal{T}}$ and shortcutting all broken paths between the remaining nodes. In particular, the rule demands that $\mathcal{H}$ is itself an optimally LiSS-stable topology in $\mathbb{T}(Ck, C, k)$. Due to this reason, we specifically investigated the LiSS-stability of head topologies in Section 4.3.

For this, we first transformed the existing characterization of LiSS-stability into a characterization using dependency graphs. Then, we led back the LiSS-stability of head topologies with unconnected dependency graphs to that of head topologies with connected dependency graphs. We identified the necessary Stability Requirements and found the Line Graph Criterion, listing sufficient conditions for optimally LiSS-stable head topologies. Additionally, we analyzed the properties of Minimum Strong Attacks on head topologies. This led to the determination of further optimally LiSS-stable head topologies and allows to restrict the search space when an exhaustive search for Minimum Strong Attacks should be necessary. The results greatly increased the number of head topologies that are available in the construction of ruled-based optimally LiSS-stable distribution topologies.

Having not yet been able to efficiently identify all optimally LiSS-stable topologies in a class $\mathbb{T}(n, C, k)$, in Section 4.4 we investigated the complexity of the LiSS-Stability Decision Problem. Here, we presented a proof that this problem is **coNP**-complete. When the input is restricted to head topologies, the complexity remained unknown. However, the class of head topologies for which Section 4.3 did not yield a way to solve the problem in **P** is characterized by quite specific dependency graphs. This fact could be the base for future results.

If $\mathbf{P} \neq \mathbf{NP}$, we saw that we have to be content with the identification of as large as possible subclasses of the optimally LiSS-stable topologies for which membership can be checked in polynomial time.

Possible heuristics to construct topologies from the identified subclasses using a distributed topology management system were briefly sketched in Section 4.5.

The results of the Sections 4.1.1–4.4 are published in [BBG+09, GFS11]. Furthermore, in [FGKS11], the author adapted them to approximate optimally LiSS-stable topologies in environments where peers may decide to receive only a subset of the available stripes. This is not allowed in the model this thesis is based on, but is possible in peer-to-peer-based IPTV systems, where the source is distributing multiple streams.

When studying the Cluster Topologies and the rule-based topologies identified in this chapter, we see that none of them impose particularly high bandwidth demands on the individual participating peers.

In fact, the requirements on the capacities of peers are of a global nature. In particular, to form Cluster Topologies from a *non-empty* class $\mathbb{T}(n, \mathrm{c}, k)$ with $\mathrm{c}(s) = Ck$, the requirements on the peer capacities stem from the necessity to form Ck inner-node-disjoint trees. This means that there must be a partition $V_1, \ldots, V_{Ck}$ of V, such that it holds that $\forall i \in [Ck]\colon \sum_{v \in V_i} \mathrm{c}(v) \geq n - 1$.

The requirements to form rule-based topologies in $\mathbb{T}(n, \mathrm{c}, k)$ are even less restrictive. Here, no such strict node partition is necessary and non-heads *may* forward in more than one stripe. However, it must be possible to adhere to the Strictly-Not-Too-Many-

Successors rule which upper-bounds the successor number of non-heads to $\delta_{C_k}^{C,k}$. As we observed in Equation (4.43), this requires a certain minimum bandwidth capacity of head nodes.

Summarizing, we can state that both identified classes of optimally LiSS-stable topologies are applicable in a wide range of real-world situations. Especially the rule-based topologies are attractive, since they provide great flexibility considering the connections and successors of non-heads. Additionally, a high number of possible head topologies is available. Furthermore, Section 5.4 will show that the huge similarities between the successor sets of the heads of Cluster Topologies render them highly vulnerable when evaluating the LoSS-damage of attacks. Since the rule-based topologies do not depend on a node clustering, they are more compatible with the requirements on topologies minimizing maximum LoSS-damage. The identification of such requirements is the topic of Chapter 5.

5. LoSS-Stability, Forward-Stability and Intersections of Successor Sets

After the successful identification of construction principles for optimally LiSS-stable topologies in Chapter 4, we will now shift our focus to study topologies that minimize, for all possible attack sizes and parameters, the maximum possible damage of an attack when using the LoSS-damage measure.

In a first step, in Section 5.1 we analyze basic properties that such topologies have to exhibit. Then we show in Section 5.2, that the LoSS-damage function can be decomposed into two superimposed types of damage, one of which – the *forward-damage* – dominates the whole function in most relevant cases. For reasons of a simplified analysis, in Section 5.3, we then resort to the study of topologies minimizing maximum forward-damage. These topologies are called *forward-stable*. Again, we identify basic necessary properties of such topologies (Subsection 5.3.1). This enables us to examine the stability of topologies adhering to these properties by using a matrix representation of their heads' forward successor sets (Subsection 5.3.2). Studying these representations allows to identify strong connections between forward-stable topologies and Design resp. Coding Theory. The latter are the topic of Subsection 5.3.3. The results cumulate in Subsection 5.3.4, where we will see that constructing forward-stable topologies involves finding solutions for long-standing open problems from Design and Coding Theory. Section 5.4 summarizes the results of the whole chapter. Furthermore, it lists unresolved research questions and recommends next steps for the research on LoSS- and forward-stable topologies.

5.1. The Problem of Finding Optimally LoSS-Stable Topologies

Let us at first review the definition of the LoSS-damage measure (see Section 3.1 for a thorough introduction). For a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, a set of nodes $X \subseteq V$ and node $v \in V$, the function $\mathrm{inc}_X(v) = |\{T_i \in \mathcal{T} \mid v \notin \mathrm{succ}_i(X)\}|$ counts the number of $s \to v$ paths that would remain in $\mathcal{T}$ after the removal of X. Given a threshold $z \in [k]$, we then defined the *Lost Service Set under Multiple Description Coding* $L^{\mathrm{MDC}}_{X,z} = \{v \in V \mid \mathrm{inc}_X(v) \leq k - z\}$ as the set of nodes that have lost at least z paths. Given X and z, the cardinality of $L^{\mathrm{MDC}}_{X,z}$ defines the LoSS-damage function $\mathrm{b}^{\mathcal{T}}(X, z) := |L^{\mathrm{MDC}}_{X,z}|$.

In Section 3.2.4, we have already studied the complexity and approximability of the LoSS problem, i.e., given $\mathcal{T}$, z and t, find an attack X of minimum cardinality such

that LoSS-damage of $\mathrm{b}^{\mathcal{T}}(X, z) \geq t$ is achieved. Now, we take an opposite position. We aim at finding topologies minimizing the maximum LoSS-damage that an attacker with complete topology knowledge but limited attack resources is able to achieve. For a justification of this worst-case approach see Section 4.1.1.

Similar to Definition 4.1.1, we define the concept of an *optimally* LoSS*-stable topology*.

Definition 5.1.1 ***Optimally* LoSS*-stable Topology***
For $n, C, k \in \mathbb{N}$ with $Ck \leq n$, a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is called *optimally* LoSS*-stable*, if it satisfies

$$\forall x \in [n], \forall z \in [k], \forall \mathcal{C} \in \mathbb{T}(n, C, k) \colon \max_{X \subseteq V, |X| = x} \mathrm{b}^{\mathcal{T}}(X, z) \leq \max_{X \subseteq V, |X| = x} \mathrm{b}^{\mathcal{C}}(X, z).$$

It is not hard to see that the optimally LoSS-stable topologies are equivalently described by a seemingly weaker condition.

Corollary 5.1.2
A topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is optimally LoSS-stable if and only if it holds that

$$\forall x \in [Ck], \forall z \in [k], \forall \mathcal{C} \in \mathbb{T}(n, C, k) \colon \max_{X \subseteq V, |X| = x} \mathrm{b}^{\mathcal{T}}(X, z) \leq \max_{X \subseteq V, |X| = x} \mathrm{b}^{\mathcal{C}}(X, z).$$

Proof. "Only-If": This follows from Definition 5.1.1.

"If": For $x \in [Ck, n]$, it holds that

$$\forall \mathcal{C} \in \mathbb{T}(n, C, k), \forall z \in [k] \colon \max_{X \subseteq V, |X| = x} \mathrm{b}^{\mathcal{T}}(X, z) = n, \tag{5.1}$$

since it is possible to choose an x-node attack X with $H^{\mathcal{T}} \subseteq X$, resulting in the disturbance of *all* source-peer paths of $\mathcal{C}$. This also implies a LoSS-damage of *all* peers. Hence, restricting the definition to the requirements defined by $x \in [Ck]$ will not exclude an optimally LoSS-stable topology. □

Furthermore, we formalize the problem of finding optimally LoSS-stable topologies.

Definition 5.1.3 ***Optimally* LoSS*-stable Topology Formation Problem***
Given $n, C, k \in \mathbb{N}$ with $n \geq Ck$, the *Optimally* LoSS*-stable Topology Formation Problem* consists in finding an optimally LoSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ or determining that none exist.

The existence of an optimally LoSS-stable topology for every set of parameters n, C, k is far from clear. If no optimally LoSS-stable topology exists, topology optimiziation should try to achieve optimality at least for small values of the attack parameters x and z, since these will be the combinations most frequently observed in practical situations.

Although optimally LoSS-stable topologies will have specific demands not seen for LiSS-stable topologies, optimizing for LiSS-stability is a way to upper-bound possible LoSS-damage.

Lemma 5.1.4
For a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ and $z \in [k]$, LiSS- and LoSS-damage of an attack $X \subseteq V$ have the following relation:

$$a^{\mathcal{T}}(X) \geq z \cdot \mathrm{b}^{\mathcal{T}}(X, z) + (k - z) \cdot |X|.$$

Proof. By definition, each of the $\mathrm{b}^{\mathcal{T}}(X, z)$ nodes in $L^{\mathrm{MDC}}_{X,z}$ must have lost at least z paths from the source, each of which is counted also as LiSS-damage. Furthermore, the attacked nodes X have lost *all* k of their paths. Since $X \subseteq L^{\mathrm{MDC}}_{X,z}$, we may only add $k - z$ for each one. □

However, optimally LiSS-stable topologies are not necessarily optimally LoSS-stable. Indeed, by enforcing high similarities between the nodes' successor sets, the optimally LiSS-stable Cluster Topologies of Section 4.1.3 can suffer very strong LoSS-attacks when compared with the *forward-stable* topologies to be introduced in Section 5.3.

From the definition of optimally LoSS-stable topologies, we can deduce a number of necessary conditions, which, all but the second, already proved to be necessary for optimal LiSS-stability.

Lemma 5.1.5
For a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, the following conditions are necessary to be optimally LoSS-stable:

1. $\forall I \subseteq [k]\colon \left|\bigcup_{i \in I} H_i^{\mathcal{T}}\right| = C \cdot |I|$
2. $\forall v \in V, \forall i, j \in [k], i \neq j\colon \mathrm{succ}_i^{\mathcal{T}}(v) \cap \mathrm{succ}_j^{\mathcal{T}}(v) = \{v\}$
3. $\forall v \in V\colon a^{\mathcal{T}}(v) \leq \left\lceil \frac{n}{C} \right\rceil + k - 1 = \delta_1^{C,k}$

Proof. We compare $\mathcal{T}$ with a Cluster Topology $\mathcal{C} \in \mathbb{T}(n, C, k)$ of depth two (cmp. Section 4.1.3). In particular, we show that the lack of each of the conditions given above allows attacks on $\mathcal{T}$ that achieve more LoSS-damage than is achievable by corresponding attacks on $\mathcal{C}$.

The Cluster Topology $\mathcal{C}$ was shown to exist in $\mathbb{T}(n, C, k)$, is optimally LiSS-stable, has $|H^{\mathcal{C}}| = Ck$, C heads per stripe, and satisfies $\forall i \in [k], \forall h \in H_i^{\mathcal{C}}\colon |\mathrm{succ}_i(h)| \leq \left\lceil \frac{n}{C} \right\rceil$. Each $v \in V \setminus H^{\mathcal{C}}$ has $|\mathrm{succ}(v)| = k$. See Figure 5.1 for an example of such a topology.

1. Assume $\exists I \subseteq [k]\colon \left|\bigcup_{i \in I} H_i^{\mathcal{T}}\right| \neq C \cdot |I|$. W.l.o.g. assume $\left|\bigcup_{i \in I} H_i^{\mathcal{T}}\right| < C \cdot |I|$ (otherwise set $I := [k] \setminus I$ since, due to limited source capacity, there are at most Ck heads in all stripes). For $z := |I|$, an attack $X := \bigcup_{i \in I} H_i^{\mathcal{T}}$ achieves $\mathrm{b}^{\mathcal{T}}(X, z) = n$.

   We show that there is no attack of less than Cz nodes achieving this amount of damage on $\mathcal{C}$. Since $\mathcal{C}$ is optimally LiSS-stable, an attack Y with $|Y| \leq Cz - 1$ nodes cannot reach LiSS-damage necessary due to Lemma 5.1.4 (we use that the

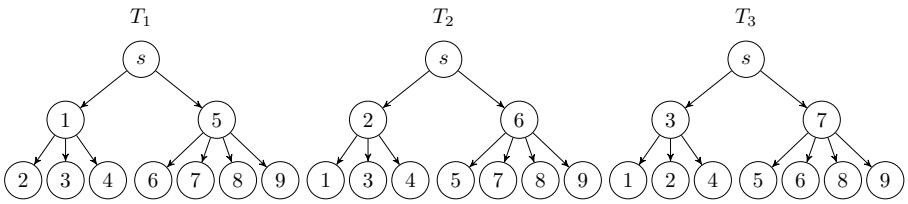


Figure 5.1.: A Cluster Topology $\mathcal{C} \in \mathbb{T}(9, 2, 3)$ as in the proof of Lemma 5.1.5.

definition of $\mathbb{T}(n, C, k)$ guarantees $\lfloor n/C \rfloor \geq k$):

$$a^{\mathcal{C}}(Y) \leq \sum_{i=1}^{Cz-1} \delta_i^{C,k} = zn + (k-z)Cz - \delta_{Cz}^{C,k} \tag{5.2}$$

$$\leq zn + (k-z)(Cz-2) - 1 < zn + (k-z)(Cz-1) \tag{5.3}$$

Thus, $X = \bigcup_{i \in I} H_j^{\mathcal{T}}$ creates more damage on $\mathcal{T}$ than any attack of $|X|$ nodes on $\mathcal{C}$. This contradicts that $\mathcal{T}$ is optimally LoSS-stable.

2. If $k = 1$, every topology in $\mathbb{T}(n, C, k)$ has Property 2. Hence, assume $k > 1$ and $\exists v, w \in V$ with $v \neq w$ such that $w \in \mathrm{succ}_i^{\mathcal{T}}(v) \cap \mathrm{succ}_j^{\mathcal{T}}(v)$ for distinct stripes i, j. Cluster Topology $\mathcal{C}$ has Property 2, so every attack with $z = 2$, $x = 1$ will lead to a damage of 1, just disabling the attacked node itself. However, in $\mathcal{T}$ we have $\mathrm{b}^{\mathcal{T}}(\{v\}, z) \geq 2$, since $\{v, w\} \subseteq L^{\mathrm{MDC}}_{\{v\},z}$. Consequently, $\mathcal{T}$ is not optimally LoSS-stable.

3. Assume $\exists v \in V \colon a^{\mathcal{T}}(v) > \lceil \frac{n}{C} \rceil + k - 1$. We set $z := 1$ and study the damage of single-node-attacks on both topologies.

   A head $h \in H^{\mathcal{C}}$ has $a^{\mathcal{C}}(h) \leq \lceil \frac{n}{C} \rceil + k - 1$ and a node $w \in V \setminus H^{\mathcal{C}}$ has $a^{\mathcal{C}}(w) = k$. By Lemma 5.1.4, it holds that $\forall w \in V \colon b^{\mathcal{C}}(\{w\}, 1) \leq \lceil \frac{n}{C} \rceil$.

   For $\mathcal{T}$, we can assume Property 2, otherwise it would be unstable. Hence, for every node $w \in V$ and two stripes $i, j \in [k], i \neq j$, we have $\mathrm{succ}_i^{\mathcal{T}}(w) \cap \mathrm{succ}_j^{\mathcal{T}}(w) = \{w\}$. Since $z = 1$, we can write

$$\mathrm{b}^{\mathcal{T}}(\{w\}, 1) = \left| \bigcup_{i \in [k]} \mathrm{succ}_i^{\mathcal{T}}(w) \right| = \left(\sum_{i \in [k]} |\mathrm{succ}_i^{\mathcal{T}}(w)| \right) - (k-1) = a^{\mathcal{T}}(w) - (k-1). \tag{5.4}$$

   This leads to $b^{\mathcal{T}}(\{v\}, 1) > \lceil \frac{n}{C} \rceil$, so attack $\{v\}$ generates more damage on $\mathcal{T}$ than any sinlge-node attack on $\mathcal{C}$. Hence, $\mathcal{T}$ is not optimally LoSS-stable.

□

Note that Property 1 does not follow from Property 3, as the topology in Figure 5.2 illustrates.

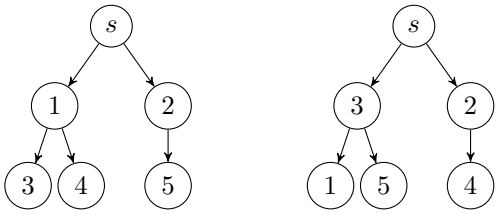


Figure 5.2.: Example topology $\mathcal{T} \in \mathbb{T}(5, 2, 2)$ violating Property 1 but satisfying Property 3 of Lemma 5.1.5.

The following result indicates that the topology heads and their successor sets play a crucial role for LoSS-stability.

Lemma 5.1.6
For each $\mathcal{T} \in \mathbb{T}(n, C, k)$ there is $\mathcal{C} \in \mathbb{T}(n, C, k)$ with depth 2, such that

$$\forall X \subseteq V, \forall z \in [k] \colon \mathrm{b}^{\mathcal{C}}(X, z) \leq \mathrm{b}^{\mathcal{T}}(X, z).$$

Proof. Construct $\mathcal{C}$ from $\mathcal{T}$ by making, in each stripe $i \in [k]$, each non-head v a child of its head $\mathrm{pred}_i^{\mathcal{T}}(v) \cap H_i^{\mathcal{T}}$ from $\mathcal{T}$. In particular, set

$$\forall v \in V, \forall i \in [k] \colon \mathrm{parent}_i^{\mathcal{C}}(v) := \begin{cases} \{s\} & , \text{ if } v \in H_i^{\mathcal{T}} \\ \mathrm{pred}_i^{\mathcal{T}}(v) \cap H_i^{\mathcal{T}} & , \text{ otherwise.} \end{cases} \tag{5.5}$$

For each node $v \in V$, there is an $s \to v$ path in each stripe of $\mathcal{C}$, since $\mathcal{T} \in \mathbb{T}(n, C, k)$ and thus $\forall i \in [k] \colon \mathrm{succ}^{\mathcal{T}}(H_i^{\mathcal{T}}) = V$. It holds that $\forall i \in [k], \forall X \subseteq V \colon \mathrm{succ}_i^{\mathcal{C}}(X) \subseteq \mathrm{succ}_i^{\mathcal{T}}(X)$, leading to $\forall v \in V \colon \mathrm{inc}_X^{\mathcal{C}}(v) \geq \mathrm{inc}_X^{\mathcal{T}}(v)$ and $\forall X \subseteq V, \forall z \in [k] \colon \mathrm{b}^{\mathcal{C}}(X, z) \leq \mathrm{b}^{\mathcal{T}}(X, z)$. □

Applying Lemma 5.1.6 to an optimally LoSS-stable topology, we obtain the following Corollary.

Corollary 5.1.7
If there is an optimally LoSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, then there is one of depth 2.

5.2. Forward-Damage and its Dominating Role

Now, we take a deeper look at the different influence factors determining the LoSS-damage achieved by an attack X on a topology $\mathcal{T}$. Since $\mathrm{succ}_i^{\mathcal{T}}(X) = \mathrm{succ}_i^{\mathcal{T}\to}(X) \cup X$, we can reformulate the definition of the Lost Service Set $L_{X,z}^{\mathrm{MDC}}$ of an attack $X \subseteq V$ on

$\mathcal{T}$ in the following way:

$$L^{\mathrm{MDC}}_{X,z} = \{v \in V \mid \mathrm{inc}_X(v) \leq k - z\} \tag{5.6}$$

$$= \left[\bigcup_{I \subseteq \{1,\dots,k\}, |I|=z} \bigcap_{i \in I} \mathrm{succ}^{\mathcal{T}}_i(X) \right] \tag{5.7}$$

$$= X \cup \left[\bigcup_{I \subseteq \{1,\dots,k\}, z=|I|} \bigcap_{i \in I} \mathrm{succ}^{\mathcal{T}\rightarrow}_i(X) \right] \tag{5.8}$$

We see that LoSS-damage is generated by the superimposition of two different aspects. The damage done to the attacked nodes X themselves will be called *direct damage* of the attack. All nodes from $L^{\mathrm{MDC}}_{X,z} \setminus X$ are successors of attacked nodes in at least z stripes and appear in Term (5.8) in at least one intersection of X's forward successor sets for a z-combination I of stripes. The damage generated by this part of $L^{\mathrm{MDC}}_{X,z}$ (not necessarily disjoint to X) will be called *forward-damage*. Due to its importance in this chapter, we will consider it as a damage function of its own:

Definition 5.2.1 ***Forward Damage Function***
For $\mathcal{T} \in \mathbb{T}(n, C, k)$, $z \in [k]$, and $X \subseteq V$, we define the *forward-damage function* as

$$\mathrm{bf}^{\mathcal{T}}(X, z) := \left| \bigcup_{I \subseteq \{1,\dots,k\}, |I|=z} \bigcap_{i \in I} \mathrm{succ}^{\mathcal{T}\rightarrow}_i(X) \right|.$$

Corollary 5.2.2
For every $\mathcal{T} \in \mathbb{T}(n, C, k)$, every $X \subseteq V$, and every $z \in [k]$, it holds that

$$\mathrm{bf}^{\mathcal{T}}(X, z) \leq \mathrm{b}^{\mathcal{T}}(X, z) \leq \mathrm{bf}^{\mathcal{T}}(X, z) + |X|.$$

Proof. This follows directly from $\mathrm{b}^{\mathcal{T}}(X, z) = |L^{\mathrm{MDC}}_{X,z}|$ and Equation (5.8). □

To find an attack maximizing the achievable LoSS-damage, an attacker has to increase forward-damage and, at the same time, aim at finding an attack set with the least intersection to the nodes suffering this forward-damage (thus he will profit from direct damage).

However, when comparing the influence of direct damage and forward-damage of an attack on the resulting LoSS-damage, we see that the effect of direct damage is only limited and stands back against the impact of forward-damage.

Theorem 5.2.3 ***Dominance of Forward Damage***
For every $\mathcal{T} \in \mathbb{T}(n, C, k), z \in [k]$, and $x \in [n]$, it holds that

$$\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z) \leq \max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z) + \min(Cz - 1, x). \tag{5.9}$$

Furthermore, for given $C, k, x \in \mathbb{N}$ and all $z \in [\min(k, x)]$, we have

$$\lim_{n \to \infty} \min_{\mathcal{T} \in \mathbb{T}(n,C,k)} \frac{\max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z)}{\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z)} = 1. \tag{5.10}$$

Proof. Let $I \subseteq [k]$ be the inidices of the z stripes of $\mathcal{T}$ with the smallest number of heads. By Lemma A.0.3, we have $\sum_{i \in I} |H_i^{\mathcal{T}}| \leq \frac{z}{k} Ck = Cz$.

If $x \geq Cz$, we can choose an attack X with $\bigcup_{i \in I} H_i^{\mathcal{T}} \subseteq X$. Then, it will hold that $V = \bigcap_{i \in I} \mathrm{succ}_i^{\mathcal{T}}(X) = \bigcap_{i \in I} \mathrm{succ}_i^{\mathcal{T}\to}(X)$ and $\mathrm{b}^{\mathcal{T}}(X, z) = \mathrm{bf}^{\mathcal{T}}(X, z) = n$. Hence, maximum LoSS- and forward-damage on $\mathcal{T}$ can only differ for $x < Cz$. Applying Corollary 5.2.2, we obtain Inequality (5.9).

Now let $\mathcal{T}$ be an arbitrary topology from $\mathbb{T}(n, C, k)$ and let values $x \in [Ck]$ and $z \in [\min(k, x)]$ be given. We show that $\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z) \geq \frac{n}{C^z}$. For this, let I be defined as above and let O be the set of all attacks containing *exactly* one head from each of the z stripes I. We have $|O| \leq \prod_{i \in I} |H_i^{\mathcal{T}}|$ with equality if no node is head in two of the stripes. Due to Lemma A.0.2, we must have $|O| \leq C^z$. For each $v \in V$, the set O must contain the set $\bigcup_{i \in I} \mathrm{pred}_i^{\mathcal{T}}(v) \cap H_i^{\mathcal{T}}$, leading to $\sum_{X \in O} \mathrm{b}^{\mathcal{T}}(X, z) \geq n$. Hence, the average forward-damage of the attacks in O is at least $\frac{n}{C^z}$ and there is $X \in O$ with $|X| \leq z$ and $\mathrm{b}^{\mathcal{T}}(X, z) \geq \frac{n}{C^z}$. Since a superset of X cannot generate less damage, it also holds that $\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z) \geq \frac{n}{C^z}$.

Due to Corollary 5.2.2 and Equation (5.9), we have

$$\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z) - Cz \leq \max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z) \leq \max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z). \tag{5.11}$$

This leads to

$$1 - \frac{Cz}{\left(\frac{n}{C^z}\right)} \leq \frac{\max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z)}{\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z)} \leq 1. \tag{5.12}$$

Hence, we obtain

$$1 = \lim_{n \to \infty} \left(1 - \frac{Cz}{\left(\frac{n}{C^z}\right)}\right) \leq \lim_{n \to \infty} \min_{\mathcal{T} \in \mathbb{T}(n,C,k)} \frac{\max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z)}{\max_{X \subseteq V, |X|=x} \mathrm{b}^{\mathcal{T}}(X, z)} \leq 1. \tag{5.13}$$

□

We see that in real-life distribution topologies, where we have $n \gg Ck$, the *LoSS damage of optimal attacks will be dominated by forward-damage*. Due to this fact, the *intersection structure* of the peers' forward successor sets – which determines forward-damage – plays a key role for the construction of LoSS-stable distribution topologies.

Topologies minimizing maximum forward-damage will be called forward-stable. Their definition builds on sets of restricted attacks.

Definition 5.2.4 ***The Set $\chi(\mathcal{T},t)$ of t-Restricted Attacks***
For a topology $\mathcal{T} \in \mathbb{T}(n,C,k)$ and $t \in [k]$, the set

$$\chi(\mathcal{T},t) := \left\{ X \subseteq V \,\middle|\, \exists I \subseteq [k] \wedge |I| = t \wedge \forall v \in X \colon \left(\bigcup_{i \in I} \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \neq \emptyset \vee \bigcup_{i \in [k]} \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) = \emptyset \right) \right\}$$

is called the *set of t-restricted attacks on $\mathcal{T}$*.

For each $X \in \chi(\mathcal{T},t)$, there is a set I of t stripes such that each $v \in X$ either has forward successors in at least one of the stripes I, or it has no forward successors at all. Consequently, if topology $\mathcal{T}$ has inner-node disjoint stripe trees, $\chi(\mathcal{T},t)$ is the set of all attacks containing inner-nodes from *at most* t stripes and an arbitrary number of nodes that are leaf in all stripes.

Definition 5.2.5 ***(t-)Forward-Stable Topology***
A topology $\mathcal{T} \in \mathbb{T}(n,C,k)$ is called *t-forward-stable*, if it holds that

$$\forall x \in [n], \forall z \in [k], \forall \mathcal{C} \in \mathbb{T}(n,C,k) \colon \max_{\substack{X \in \chi(\mathcal{T},t) \\ |X|=x}} \mathrm{bf}^{\mathcal{T}}(X,z) \leq \max_{\substack{X \in \chi(\mathcal{C},t) \\ |X|=x}} \mathrm{bf}^{\mathcal{C}}(X,z).$$

If $\mathcal{T}$ is t-forward-stable for all $t \in [k]$, it is called *forward-stable*.

Informally, a topology $\mathcal{T} \in \mathbb{T}(n,C,k)$ is t-forward-stable, if it minimizes the maximum possible forward damage that is achievable by t-restricted attacks (for all attack cardinalities and thresholds z). Forward-stable topologies are t-forward-stable for all possible values of t. The definition of $\chi(\mathcal{T},t)$ leads to $\chi(\mathcal{T},t) \subseteq \chi(\mathcal{T},t+1)$ for all $t \in [k-1]$. Furthermore, it holds that $\chi(\mathcal{T},k) = \mathcal{P}(V)$. Thus, a forward-stable topology must – similar to a LiSS- or LoSS-stable topology – be stable against worst-case attacks of all possible cardinalities and parameters.

Again, we define a corresponding topology formation problem.

Definition 5.2.6 ***(Restricted) Forward-Stable Topology Formation Problem***
The *(Restricted) Forward-Stable Topology Formation Problem* for input parameters n, C, k (and t) consists in finding a (t-)forward-stable topology $\mathcal{T} \in \mathbb{T}(n,C,k)$ or determining that none exists.

In the following, we will see that forward-stable topologies exist. When such a solution is found and output, it will have at least nk edges. This is exponential in the binary representation of the input variables. Hence, as with other topology formation problems, any algorithm solving the (Restricted) Forward-Stable Topology Formation Problem must have at least pseudopolynomial runtime.

Motivation for the Analysis of Forward-Stable Topologies In the remaining sections of this chapter, we will focus on studying the properties and the construction of forward-stable topologies. By omitting, from the LoSS-damage, the damage of attacked nodes that are not hit by forward-damage, we gain a considerably simplified analysis. In particular, we do not have to handle the superimposition of the effects of a) the intersection structure of the topology's forward successor sets and b) the choice of actual nodes these forward successor sets are associated with.

In the long term, this will allow us to identify many unstable topologies based on simple requirements and to analyze the remaining topologies based on a compact matrix representation. Using this representation, we can connect our problems with a large number of results from Design and Coding Theory. In particular, we will show that the construction of forward-stable topologies is at least as hard as solving long-standing open problems in these areas.

5.3. Constructing Forward-Stable Topologies

The following Subsection 5.3.1 identifies basic properties of forward-stable topologies. In particular, its results allow us to introduce a matrix representation of the heads' successor sets in Subsection 5.3.2. Such a matrix characterizes the forward-stability of a distribution topology and we show that it has to be a so-called Orthogonal Array or a special Packing Array. If $n \leq C^k$, the rows of these matrices can be seen as an error-correcting code. Therefore, Subsection 5.3.3 gives a review on related coding-theoretical results and their possible application in the study of forward-stable topologies. Finally, Subsection 5.3.4 investigates the existence conditions of the necessary matrices and connects the (Restricted) Forward-Stable Topology Formation Problem with open problems in Design and Coding Theory.

5.3.1. Basic Properties of Forward-Stable Topologies

In Lemma 5.3.2, we identify necessary conditions on t-forward-stable topologies. Then, we check whether these conditions conflict with the necessary conditions on optimally LoSS-stable topologies. Finally, we find out that the forward-stability of topologies adhering to the properties given in Lemma 5.3.2 depends only on the forward successor sets of their heads.

We start with a technical corollary about small attacks on t-forward-stable topologies.

Corollary 5.3.1
A t-forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ satisfies for all $x \in [t]$

$$\forall z \in [k], \forall \mathcal{C} \in \mathbb{T}(n, C, k)\colon \max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z) \leq \max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{C}}(X, z). \tag{5.14}$$

Proof. For every topology $\mathcal{C} \in \mathbb{T}(n, C, k)$ and every set $X \subseteq V$ with $|X| = x \leq t$, there is a set $I \subseteq [k]$ of cardinality at most t such that $\forall v \in X\colon I \cap \arg\max_{i \in [k]} |\mathrm{succ}_i^{\mathcal{C}\rightarrow}(v)| \neq \emptyset$.

Hence, $\chi(\mathcal{C}, t)$ contains *all* possible attacks on x nodes from V:

$$\forall \mathcal{C} \in \mathbb{T}(n, C, k) \colon \{X \subseteq V \mid |X| = x\} = \{X \in \chi(\mathcal{C}, t) \mid |X| = x\}. \tag{5.15}$$

Since $\mathcal{T}$ is t-forward-stable, it especially holds that

$$\forall z \in [k], \forall \mathcal{C} \in \mathbb{T}(n, C, k) : \max_{X \in \chi(\mathcal{T}, t), |X| = x} \mathrm{bf}^{\mathcal{T}}(X, z) \leq \max_{X \in \chi(\mathcal{C}, t), |X| = x} \mathrm{bf}^{\mathcal{C}}(X, z). \tag{5.16}$$

Due to Line (5.15), the Inequalities (5.16) and (5.14) are equivalent. □

Lemma 5.3.2
A t-forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ with $t \in [k]$ has the following properties:

1. $\forall v \in V \colon |\{i \in [k] \mid \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \neq \emptyset\}| \leq 1$
2. $\forall v \in V \colon |\bigcup_{i \in [k]} \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v)| \leq \lceil \frac{n}{C} \rceil$

Proof. Similar to the proof of Lemma 5.1.5, we compare $\mathcal{T}$ with a Cluster Topology $\mathcal{C} \in \mathbb{T}(n, C, k)$ of depth 2. In each stripe i of $\mathcal{C}$, every $v \in V \setminus H_i^{\mathcal{C}}$ has $|\mathrm{succ}_i^{\mathcal{C}\rightarrow}(v)| = 0$.

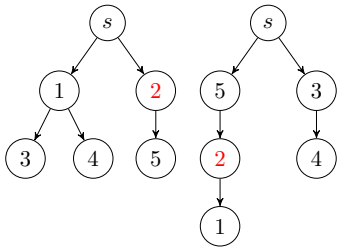
Due to Corollary 5.3.1, $\mathcal{T}$ should minimize the maximum forward-damage for attacks of cardinality 1 and all values of z. However, if $\mathcal{T}$ lacks one of the properties, we show that, for certain z, there are attacks of cardinality 1 on $\mathcal{T}$ that achieve more forward-damage than any such attack can achieve on $\mathcal{C}$:

1. Assume there is $v \in V$ and two distinct stripes i, j, such that $\mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \neq \emptyset$ and $\mathrm{succ}_j^{\mathcal{T}\rightarrow}(v) \neq \emptyset$. Then, we must have $v \in \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \cap \mathrm{succ}_j^{\mathcal{T}\rightarrow}(v)$. In contrast, for all $w \in V$ there is no pair i, j of distinct stripes of $\mathcal{C}$ such that $\mathrm{succ}_i^{\mathcal{C}\rightarrow}(w) \cap \mathrm{succ}_j^{\mathcal{C}\rightarrow}(w) \neq \emptyset$. We obtain $\max_{X \subseteq V, |X|=1} \mathrm{bf}^{\mathcal{T}}(X, 2) \geq 1$, whereas it holds that $\max_{Y \subseteq V, |Y|=1} \mathrm{bf}^{\mathcal{C}}(Y, 2) = 0$. Hence, $\mathcal{T}$ is not t-forward-stable.

2. Assume that $\exists v \in V \colon |\bigcup_{i \in [k]} \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v)| > \lceil \frac{n}{C} \rceil$. The maximum forward-damage of an attack with $z = 1, x = 1$ on a topology $\mathcal{D} \in \mathbb{T}(n, C, k)$ equals $\max_{v \in V} |\bigcup_{i \in [k]} \mathrm{succ}_i^{\mathcal{D}\rightarrow}(v)|$. In $\mathcal{C}$, this value is $\lceil \frac{n}{C} \rceil$, whereas, due to v, it is higher in $\mathcal{T}$. Hence, $\mathcal{T}$ is not t-forward-stable.

□

Note, that the first property prohibits nodes forwarding or being head in multiple stripes. We will call this the *one-stripe-only property*. Both properties together induce all the properties necessary for LoSS-stable topologies stated in Lemma 5.1.5. However, with the one-stripe-only property, the requirements for forward-stability are more strict. Example 5.3.3 demonstrates, that this is not necessary for optimal LoSS-stability.



(a) $\mathcal{T} \in \mathbb{T}(5, 2, 2)$

$\lvert X\rvert$	X	$\mathrm{b}(X, 1)$	X	$\mathrm{b}(X, 2)$
1	$\{1\}, \{2\}, \{5\}$	3	$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$	1
2	$\{1, 2\}, \{3, 5\}$	5	$\{1, 3\}$	3
3	–	–	$\{1, 2, 3\}, \{1, 3, 5\}$	4
4	–	–	$\{1, 2, 3, 5\}$	5

(b) Attacks of maximum LoSS-damage on $\mathcal{T}$.

Figure 5.3.: Distribution topology for Example 5.3.3. Head 2 forwarding in both stripes.

Example 5.3.3 ***LoSS-stable Topology Without One-Stripe-Only***
The maximum LoSS-damage on the topology depicted in Figure 5.3 equals that on an optimally LoSS-stable topology $\mathcal{T}^* \in \mathbb{T}(5, 2, 2)$:

$z = 1$: Topology $\mathcal{T}^*$ must have a stripe i with head $h \in H_i^{\mathcal{T}^*}$, such that $\mathrm{succ}_i^{\mathcal{T}^*}(h) \geq \lceil \frac{5}{2} \rceil = 3$. Thus, since $\mathrm{b}^{\mathcal{T}^*}(X, 1) = |\bigcup_{j \in [k]} \mathrm{succ}_j^{\mathcal{T}^*}(X)|$, we have $\max_{v \in V} \mathrm{b}^{\mathcal{T}^*}(v, 1) \geq 3$. Furthermore, attacking both heads $H_i^{\mathcal{T}^*}$ damages all 5 nodes. The LoSS-damage cannot be increased for larger attacks.

$z = 2$: Every non-empty attack on $\mathcal{T}^*$ will achieve a damage of at least 1 and every attack on the 4 heads $H^{\mathcal{T}^*}$ leads to LoSS-damage of $n = 5$. For attacks of size 2, $\mathcal{T}^*$ must have an attack targeting both heads of the node in $V \setminus H^{\mathcal{T}^*}$, leading to a damage of 3. Enlarging this attack set by another head results in LoSS-damage of 4.

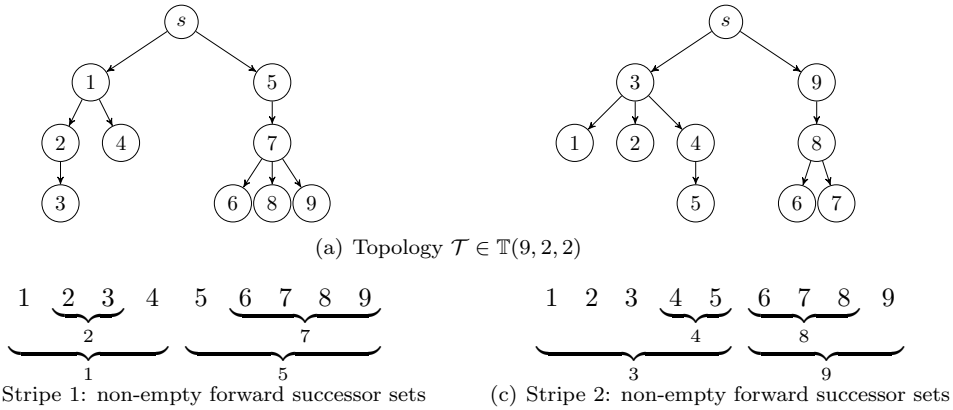
However, it is safe to dictate the one-stripe-only property, since, although not all optimally LoSS-stable topologies possess this property, we can always find one adhering to it if any optimally LoSS-stable topology exists.

Lemma 5.3.4
If there is an optimally LoSS-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, then there is an optimally LoSS-stable topology $\mathcal{C} \in \mathbb{T}(n, C, k)$ that has all properties listed in Lemma 5.3.2.

Proof. At first, we show this for Property 1, the one-stripe-only property. By Corollary 5.1.7, there is an optimally LoSS-stable topology $\mathcal{C} \in \mathbb{T}(n, C, k)$ with depth 2. This topology must have the properties of Lemma 5.1.5. Then, only the topology heads will forward the stream and each head will forward just in the (one) stripe it is head of. Thus, $\mathcal{C}$ adheres to the one-stripe-only property.

Now, Property 2 of Lemma 5.3.2 follows from the combination of the one-stripe-only property and Property 3 of Lemma 5.1.5. □

Topologies satisfying the conditions of Lemma 5.3.2 have the following property.



(a) Topology $\mathcal{T} \in \mathbb{T}(9, 2, 2)$

(b) Stripe 1: non-empty forward successor sets

(c) Stripe 2: non-empty forward successor sets

Figure 5.4.: A topology adhering to the properties of Lemma 5.3.2 and the laminar families of non-empty forward successor sets in each of its stripes.

Lemma 5.3.5
Let $\mathcal{T} \in \mathbb{T}(n, C, k)$ have the properties given in Lemma 5.3.2. For all $z \in [k]$ and each $X \subseteq V$, there is $Y \subseteq H^{\mathcal{T}}$ with $\mathrm{bf}^{\mathcal{T}}(Y, z) \geq \mathrm{bf}^{\mathcal{T}}(X, z)$ and $|Y| = \min(|X|, Cz)$.

Proof. Due to the one-stripe-only property, the sets $V_i := \{v \in V \mid \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \neq 0\}$ for $i \in [k]$ together with the set $V_0 := V \setminus \bigcup_{i \in [k]} V_i$ form a partition of V.

Furthermore, since each stripe $T_i \in \mathcal{T}$ is a tree, the forward successor set of each node $v \in V$ is is a proper subset of the forward successor set of each predecessor $\mathrm{pred}_i^{\mathcal{T}\rightarrow}(v)$. Hence, for each $i \in [k]$, the set $\{\mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \mid v \in V_i\}$ is a *laminar family of sets*, i.e., for every two sets A, B from this family, $A \cap B$ equals either A, B, or $\emptyset$. In $\{\mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \mid v \in V_i\}$, the forward successor sets of the heads $H_i^{\mathcal{T}}$ are the only sets that are not subsets of others (cmp. Figure 5.4).

Now, let $X \subseteq V$ be an arbitrary attack on $\mathcal{T}$. If $|X| \geq Cz$, then *all* nodes can be isolated by attacking the (at most) Cz heads of z stripes with the smallest number of heads.

Otherwise, drop nodes with only empty forward successor sets from X, i.e., set $X := X \setminus V_0$, and let

$$Y' := \{h \in H^{\mathcal{T}} \mid \exists i \in [k], \exists v \in V_i \cap X \colon \mathrm{succ}_i^{\mathcal{T}\rightarrow}(v) \subseteq \mathrm{succ}_i^{\mathcal{T}\rightarrow}(h)\}. \tag{5.17}$$

Due to the node partition and set laminarity, it holds that $|Y'| \leq |X|$. Furthermore, we have $\forall i \in [k] \colon \mathrm{succ}_i^{\mathcal{T}\rightarrow}(X) \subseteq \mathrm{succ}_i^{\mathcal{T}\rightarrow}(Y')$ and therefore $\mathrm{bf}^{\mathcal{T}}(X, z) \leq \mathrm{bf}^{\mathcal{T}}(Y', z)$ for all $z \in [k]$. No superset $Y \subseteq H^{\mathcal{T}}$ with $Y' \subseteq Y$ and $|Y| = |X|$ can create less forward-damage. □

We see that the forward-stability of topologies with the properties given in Lemma 5.3.2 depends only on the forward successor sets of their heads.

Corollary 5.3.6
Let $\mathcal{T} \in \mathbb{T}(n, C, k)$ have the properties given in Lemma 5.3.2. For all $z \in [k]$ and all $x \in [Ck]$, the value

$$\max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z)$$

is completely determined by the forward successor sets of the heads $H^{\mathcal{T}}$.

Proof. Due to Lemma 5.3.5, there is $Y \in \arg\max_{X \subseteq V, |X|=x} \mathrm{bf}^{\mathcal{T}}(X, z)$ with $Y \subseteq H^{\mathcal{T}}$ for each $z \in [k]$ and $x \in [Ck]$. Its forward-damage is determined only by the forward successor sets of the nodes Y. □

5.3.2. Head Forward Successor Sets and Orthogonal Arrays

Due to Corollary 5.3.6, all further study on forward-stable topologies can concentrate on the structure of the multiset $F := \{\mathrm{succ}_i^{\mathcal{T}\rightarrow}(h) \neq \emptyset \mid h \in H^{\mathcal{T}}, i \in [k]\}$ of non-empty forward successor sets of heads. In a topology $\mathcal{T}$ with the properties of Lemma 5.3.2, each head has a non-empty forward successor set in exactly one stripe and it holds that $|F| = |H^{\mathcal{T}}| = Ck$. Additionally, the definition of forward successor sets guarantees that for each $i \in [k]$ the set $F_i := \{\mathrm{succ}_i^{\mathcal{T}\rightarrow}(h) \mid h \in H_i^{\mathcal{T}}\}$ is a partition of V into $|H_i^{\mathcal{T}}|$ sets. For the following analysis, we will use a matrix-based representation of F. For its definition, recall that the topologies $\mathbb{T}(n, C, k)$ are defined on the node set $V = [n]$.

Definition 5.3.7 ***Matrix $M^{\mathcal{T}}$ of Forward Successor Sets of $H^{\mathcal{T}}$***
Let $\mathcal{T} \in \mathbb{T}(n, C, k)$ be given. Using a bijection $\sigma_i \colon H_i^{\mathcal{T}} \to [\,|H_i^{\mathcal{T}}|\,]$ per stripe $i \in [k]$, the *matrix $M^{\mathcal{T}}$ of forward successor sets of the heads $H^{\mathcal{T}}$* is an $n \times k$ matrix $M^{\mathcal{T}} = (m_{vi})$, such that for each $v \in V, i \in [k]$, and $j \in H_i^{\mathcal{T}}$, it holds that

$$m_{vi} = \sigma_i(j) \Leftrightarrow v \in \mathrm{succ}_i^{\mathcal{T}\rightarrow}(j).$$

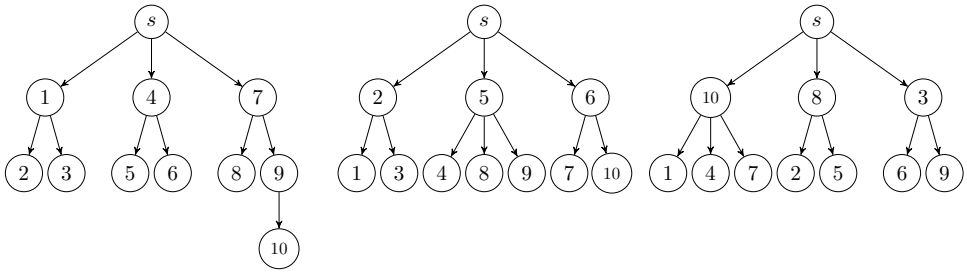
For $v \in V$, define $M^{\mathcal{T}}[v] = (m_{v1}, \ldots, m_{vk})$ as the v-th row of $M^{\mathcal{T}}$.

This matrix representation illustrates the membership relations between the elements of V and F. The i-th entry of the v-th row of $M^{\mathcal{T}}$ encodes the head supplying node v in stripe i. Its numeric value is determined by bijection σ_i. See Figure 5.5 for an example. Given the bijections σ_i, the multiset F can be completely recovered from $M^{\mathcal{T}}$ since, due to its definition, it holds that $\forall h \in H^{\mathcal{T}}, \forall i \in [k] \colon \mathrm{succ}_i^{\mathcal{T}\rightarrow}(h) = \{w \in V \mid m_{wi} = \sigma_i(h)\}$.

Reusing the bijections σ_i from $M^{\mathcal{T}}$, we can also transform each attack $X \subseteq H^{\mathcal{T}}$ into a set $\sigma(X)$ of k-dimensional vectors. In their i-th position, these vectors contain entries from $\{\sigma_i(h) \mid h \in X \cap H_i^{\mathcal{T}}\}$ if $X \cap H_i^{\mathcal{T}} \neq \emptyset$. Otherwise, this position contains value 0.

Definition 5.3.8 ***Vector Attack***
Given $\mathcal{T} \in \mathbb{T}(n, C, k)$, the matrix $M^{\mathcal{T}}$, and the corresponding bijections $\sigma_i \colon H_i^{\mathcal{T}} \to [\,|H_i^{\mathcal{T}}|\,]$ for $i \in [k]$, the *vector attack* $\sigma(X)$ for an attack $X \subseteq H^{\mathcal{T}}$ on $\mathcal{T}$ is defined as

$$\sigma(X) := \left\{\mathbf{y} \in (\{0\} \cup \mathbb{N})^k \mid \forall i \in [k] \colon ((\mathbf{y}_i = 0) \wedge (X \cap H_i^{\mathcal{T}} = \emptyset)) \vee (\sigma_i^{-1}(\mathbf{y}_i) \in X)\right\}.$$



(a) Topology $\mathcal{T} \in \mathbb{T}(10, 3, 3)$

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 2 & 2 & 3 & 3 & 2 & 2 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 \end{pmatrix}$$

(b) $M^{\mathcal{T}^T}$ for $\sigma_1(1) = \sigma_2(2) = \sigma_3(10) = 1$, $\sigma_1(4) = \sigma_2(5) = \sigma_3(8) = 2$ and $\sigma_1(7) = \sigma_2(6) = \sigma_3(3) = 3$

Figure 5.5.: A topology $\mathcal{T}$ with the properties of Lemma 5.3.2 and the transposed matrix $M^{\mathcal{T}}$ of the forward successor sets of $H^{\mathcal{T}}$.

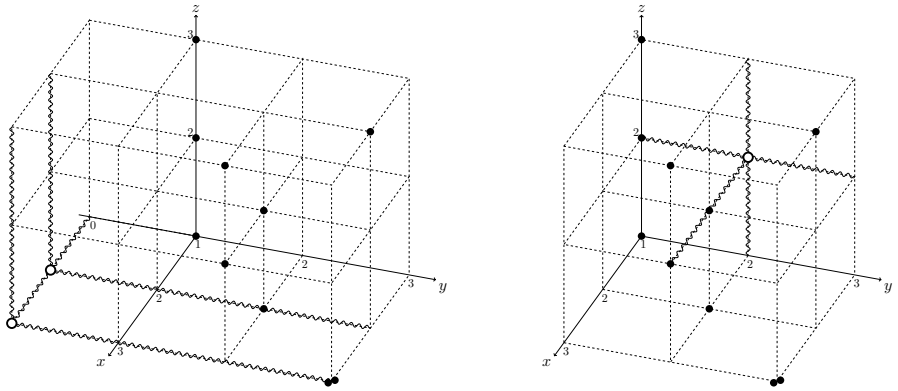
We can restate our notion of forward-damage by counting certain row vectors in $M^{\mathcal{T}}$. Using vector attacks, the forward-damage of an attack $X \subseteq H^{\mathcal{T}}$ on $\mathcal{T}$ is equivalent to

$$\begin{aligned} \mathrm{bf}^{\mathcal{T}}(X, z) &= \left| \bigcup_{I \subseteq [k], |I| = z} \bigcap_{i \in I} \mathrm{succ}_i^{\mathcal{T}\rightarrow}(X) \right| && (5.18) \\ &= \left| \left\{ v \in V \;\middle|\; \exists I \subseteq [k], |I| = z \colon v \in \bigcap_{i \in I} \mathrm{succ}_i^{\mathcal{T}\rightarrow}(X) \right\} \right| && (5.19) \\ &= \left| \left\{ v \in V \;\middle|\; \exists I \subseteq [k], |I| = z, \forall i \in I \colon \sigma_i^{-1}(m_{vi}) \in X \right\} \right| && (5.20) \\ &= \left| \left\{ v \in V \;\middle|\; \exists I \subseteq [k], |I| = z, \exists \mathbf{x} \in \sigma(X), \forall i \in I \colon \mathbf{x}_i = m_{vi} \right\} \right| && (5.21) \\ &= \left| \left\{ v \in V \;\middle|\; \exists \mathbf{x} \in \sigma(X) \colon \mathrm{d}(M[v], \mathbf{x}) \leq k - z \right\} \right|, && (5.22) \end{aligned}$$

where $\mathrm{d}(\cdot, \cdot)$ is the *Hamming Distance* between two vectors. Example 5.3.9 demonstrates this connection between vector attacks and forward-damage.

Example 5.3.9 ***Vector Attacks and Forward Damage***
Let $X = \{4, 7, 10\}$ and $Y = \{1, 5, 8\}$ be attacks on the topology $\mathcal{T}$ of Figure 5.5(a) with $M^{\mathcal{T}}$ and σ_i as given in Figure 5.5(b).

We have $\sigma(X) = \{(2, 0, 1), (3, 0, 1)\}$ and $\sigma(Y) = \{(1, 2, 2)\}$. The rows of $M^{\mathcal{T}}$ must be vectors from $[\,|H_1^{\mathcal{T}}|\,] \times [\,|H_2^{\mathcal{T}}|\,] \times [\,|H_3^{\mathcal{T}}|\,] = [3]^3$. For $z = 2$, in $[3]^3$ the vectors $\{(2, 1, 1), (2, 2, 1), (2, 3, 1), (3, 1, 1), (3, 2, 1), (3, 3, 1)\}$ have Hamming Distance ≤ 1 to $\sigma(X)$ and the vectors $\{(1, 2, 2), (2, 2, 2), (3, 2, 2), (1, 1, 2), (1, 3, 2), (1, 2, 1), (1, 2, 3)\}$ have a distance ≤ 1 to $\sigma(Y)$. Each $v \in V$ with $M^{\mathcal{T}}[v]$ in these sets suffers forward-damage. This applies to $\{4, 7, 10\}$ for X and $\{2, 5, 8\}$ for Y (visualized in Figure 5.6).



(a) Rows of $M^{\mathcal{T}}$ in distance ≤ 1 from vector attack $\sigma(X)$.

(b) Rows of $M^{\mathcal{T}}$ in distance ≤ 1 from vector attack $\sigma(Y)$.

Figure 5.6.: Illustration of Example 5.3.9. Rows of $M^{\mathcal{T}}$ black dots, vector attacks circled, neighborhood of Hamming Distance 1 snaked.

$$\begin{pmatrix}
1&1&1&1&1&1&1&1&1&2&2&2&2&2&2&2&2&2&3&3&3&3&3&3&3&3&3\\
1&1&1&2&2&2&3&3&3&1&1&1&2&2&2&3&3&3&1&1&1&2&2&2&3&3&3\\
1&2&3&1&2&3&1&2&3&1&2&3&1&2&3&1&2&3&1&2&3&1&2&3&1&2&3\\
1&2&3&2&3&1&3&1&2&2&3&1&3&1&2&1&2&3&3&1&2&1&2&3&2&3&1
\end{pmatrix}$$

(a) A transposed OA$(27, 4, 3, 3)$ of index 1.

$$\begin{pmatrix}
1&1&1&2&2&2&3&3&3&1&1&1&2&2&2&3&3&3\\
1&2&3&1&2&3&1&2&3&1&2&3&1&2&3&1&2&3\\
1&2&3&2&1&3&3&1&2&1&3&2&3&2&1&2&3&1
\end{pmatrix}$$

(b) A transposed indecomposable OA$(18, 3, 3, 2)$ of index 2 [DB02].

Figure 5.7.: Examples of Orthogonal Arrays

Now we can introduce the following concept from Design Theory [HSS99, CD06].

Definition 5.3.10 ***Orthogonal Array***
For $n, k, C \in \mathbb{N}$ and $t \in [0, k]$, an $n \times k$ matrix M with entries $m_{vi} \in [C]$ is called an *Orthogonal Array* $\mathrm{OA}(n, k, C, t)$ if in every $n \times t$ submatrix M' consisting of t complete columns of M, each $\mathbf{x} \in [C]^t$ appears exactly $\lambda := \frac{n}{C^t}$ times as a row.

We say that M has *strength* t, C *levels*, k *factors* and *index* λ.

Figure 5.7(a) shows an Orthogonal Array built using the Bush construction [Bus52]. Orthogonal Arrays have the following nice property.

Corollary 5.3.11 ***Strength of Orthogonal Arrays***
An Orthogonal Array M of strength t also has strength $t - 1$.

Proof. Each $n \times (t-1)$ submatrix M'' of M contains all but one column of an $n \times t$ submatrix M' of M. M' contains each t-tuple of symbols from $[C]$ exactly $\frac{n}{C^t}$ times as row. C of these t-tuples at a time share the same symbols in the columns of M''. Thus, each $(t-1)$-tuple of symbols will occur in $C\frac{n}{C^t} = \frac{n}{C^{t-1}}$ rows of M''. □

Clearly, reusing this corollary shows that M must be of strength $t' \in \mathbb{N}$ for all $t' \leq t$. Furthermore, we can combine two Orthogonal Arrays to obtain an Orthogonal Array with more rows.

Corollary 5.3.12 ***Concatenation of Orthogonal Arrays***
Let M_1 be an $\mathrm{OA}(n_1, k, C, t_1)$ and M_2 be an $\mathrm{OA}(n_2, k, C, t_2)$, then

$$M_3 = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$$

is an $\mathrm{OA}(n_1 + n_2, k, C, \min(t_1, t_2))$.

Proof. By Corollary 5.3.11 both M_1 and M_2 are of strength $t_3 := \min(t_1, t_2)$. Hence, each $(n_1 + n_2) \times t_3$ submatrix of t_3 columns of M_3 will contain each possible vector $\mathbf{v} \in [C]^{t_3}$ exactly $\frac{n_1}{C^{t_3}}$ times from M_1 and $\frac{n_2}{C^{t_3}}$ times from M_2. Resulting in an overall frequency of $\frac{n_1+n_2}{C^{t_3}}$. □

The reverse operation of obtaining two Orthogonal Arrays M_1 and M_2 of strength t by bipartitioning the rows of an Orthogonal Array M_3 of strength t is known as *decomposition of Orthogonal Arrays*. Since n is a multiple of C^t in every Orthogonal Array, it is important to note that *not* every Orthogonal Array of index greater one is indeed decomposable [DB02]. Figure 5.7(b) shows an example. Although this $\mathrm{OA}(18, 3, 3, 2)$ is indecomposable, there actually exist $\mathrm{OA}(9, 3, 3, 2)$. However it is not always the case that there is an $\mathrm{OA}(C^t, k, C, t)$. The problem of finding the minimum value n, such that an $\mathrm{OA}(n, k, C, t)$ exists will be among the topics of Section 5.3.4.

The following lemma shows, that each Orthogonal Array of strength at least 1 satisfies our current requirements for a use in forward-stable topologies.

Lemma 5.3.13
For every $\mathrm{OA}(n, k, C, t)$ M with $n \geq Ck$ and strength $t \geq 1$, there is a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ with $M^{\mathcal{T}} = M$ that satisfies the requirements of Lemma 5.3.2.

Proof. We construct a suitable topology $\mathcal{T}$ of depth 2. For the use as heads $H^{\mathcal{T}}$, we determine the indices of Ck suitable rows of M. For this, we construct a bipartite graph $G = \big([n] \dot\cup ([C] \times [k]), E\big)$. Its node set contains the n peers $[n] = V$ of $\mathcal{T}$ and head positions (i, j). A head position (i, j) corresponds with the role as i-th head in stripe j of $\mathcal{T}$. The edge set E satisfies

$$\{v, (i, j)\} \in E \Leftrightarrow M[v]_j = i. \tag{5.23}$$

For each $u \in [n] \dot\cup ([C] \times [k])$, define $N(u) := \{w \in [n] \dot\cup ([C] \times [k]) \mid \{u, w\} \in E\}$. Since M has k columns, each node $v \in [n]$ satisfies $|N(v)| = k$. Since M has strength

at least 1, each head position (i,j) has $|N((i,j))| = n/C$. Due to Hall's Theorem (cmp. [Die05]) there is a matching covering all head positions in G, if it holds that $\forall S \subseteq [C] \times [k]\colon |\bigcup_{u\in S} N(u)| \geq |S|$. We show that this is the case in G. For each possible subset S of head positions, there are $|S| \cdot n/C$ edges to nodes from $[n]$. Since these $|\bigcup_{u\in S} N(u)|$ nodes have $|\bigcup_{u\in S} N(u)| \cdot k$ edges in total and since $n/C \geq k$, we obtain

$$|S| \cdot \frac{n}{C} \leq \Big|\bigcup_{u\in S} N(u)\Big| \cdot k \tag{5.24}$$

$$\Rightarrow |S| \leq \Big|\bigcup_{u\in S} N(u)\Big|. \tag{5.25}$$

Hence, there is a (maximum) matching R in G that connects each head position with a unique node from $[n]$. For each $\{v,(i,j)\} \in R$, we use v as head in stripe j of $\mathcal{T}$ and define $\sigma_j(v) := i$. Furthermore, we set $\text{child}_j^{\mathcal{T}}(v) := \{u \in [n] \setminus \{v\} \mid M[u]_j = i\}$. In each stripe of the emerging topology $\mathcal{T}$, every node is either head or child of a head. The matching R guarantees that we have $|H^{\mathcal{T}}| = Ck$ and that each head forwards in only one stripe. The defined bijections σ_j with $j \in [k]$ establish $M^{\mathcal{T}} = M$. Since M is of strength at least 1, for all $j \in [k]$ each head $v \in H_j^{\mathcal{T}}$ satisfies $|\text{succ}_j^{\mathcal{T}\rightarrow}(v)| = n/C$. All other forward successor sets are empty. □

If we can choose $M^{\mathcal{T}}$ as an Orthogonal Array with high strength t, this will be very benificial for $\mathcal{T}$'s forward-stability.

Theorem 5.3.14
A topology $\mathcal{T} \in \mathbb{T}(n,C,k)$ is t-forward-stable, if it has the properties of Lemma 5.3.2 and $M^{\mathcal{T}}$ is an $\text{OA}(n,k,C,t)$.

Proof. We show that, for every attack $X \in \chi(\mathcal{T},t)$ on such a topology $\mathcal{T}$ and every topology $\mathcal{C} \in \mathbb{T}(n,C,k)$, there is an attack $Y \in \chi(\mathcal{C},t)$ with $\forall z \in [k] : \text{bf}^{\mathcal{T}}(X,z) \leq \text{bf}^{\mathcal{T}}(Y,z)$. To do this, we introduce and apply the following concepts.

Let us call $\mathbf{s} \in \mathbb{N}^k$ with $\sum_{i=1}^k \mathbf{s}_i = Ck$ a *head distribution* and define the *head vector space for* $\mathbf{s}$ as

$$\mathbb{V}(\mathbf{s}) := [\mathbf{s}_1] \times \ldots \times [\mathbf{s}_k]. \tag{5.26}$$

For every topology $\mathcal{C} \in \mathbb{T}(n,C,k)$ with $\forall i \in [k]\colon |H_i^{\mathcal{C}}| = \mathbf{s}_i$, the rows of $M^{\mathcal{C}}$ are elements of $\mathbb{V}(\mathbf{s})$. This follows from the definition of $M^{\mathcal{C}}$, since each σ_i maps to $[\,|H_i^{\mathcal{C}}|\,]$.

A vector $\mathbf{a}$ of k elements with $\forall i \in [k]\colon \mathbf{a}_i \in [0,\mathbf{s}_i]$ will be named *attack distribution* for $\mathbf{s}$. Given such an $\mathbf{a}$, all vector attacks $\sigma(X)$ for attacks $X \subseteq H^{\mathcal{C}}$ with $\forall i \in [k]\colon |X \cap H_i^{\mathcal{C}}| = \mathbf{a}_i$ are contained in the set

$$\mathbb{X}(\mathbf{a},\mathbf{s}) := \Big\{\mathfrak{X} = \mathfrak{X}_1 \times \ldots \times \mathfrak{X}_k \;\Big|\; \forall i \in [k]\colon (0 = \mathbf{a}_i \wedge \mathfrak{X}_i = \{0\}) \vee (\,0 < \mathbf{a}_i = |\mathfrak{X}_i| \wedge \mathfrak{X}_i \subseteq [\mathbf{s}_i]\,)\Big\}. \tag{5.27}$$

Note that $\mathbb{X}(\mathbf{a}, \mathbf{s})$ and $\mathbb{V}(\mathbf{s})$ will not intersect if $\mathbf{a}$ has entries of value zero. We will call all elements of $\mathbb{X}(\mathbf{a}, \mathbf{s})$ vector attacks, although it is possible that there is $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$ such that no $X \subseteq H^{\mathcal{C}}$ with $\sigma(X) = \mathfrak{X}$ exists. This case can appear if $\mathcal{C}$ has nodes that are heads in multiple stripes. An example is given by a topology with $C = 1, k = 2$ and a single, identical head in both stripes. Here, there will be no head attack with attack distribution $(1, 0)$.

For a vector attack $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$, we define the i-neighborhood $N_i(\mathfrak{X}, \mathbf{s})$ of $\mathfrak{X}$ in $\mathbb{V}(\mathbf{s})$ as

$$N_i(\mathfrak{X}, \mathbf{s}) := \{\mathbf{v} \in \mathbb{V}(\mathbf{s}) \mid i = \min_{\mathbf{x} \in \mathfrak{X}} \mathrm{d}(\mathbf{v}, \mathbf{x})\}, \tag{5.28}$$

where $\mathrm{d}(\cdot, \cdot)$ is, again, the Hamming Distance. If $\mathbf{v} \in N_i(\mathfrak{X}, \mathbf{s})$, we will say that $\mathbf{v}$ *has minimum distance i to $\mathfrak{X}$*.

Before we can continue the proof, we have to establish the following three claims about neighborhoods of vector attacks.

Claim 5.3.15
Let $\mathbf{a}$ be an attack distribution for a head distribution $\mathbf{s}$. Then it holds that

$$\forall \mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s}), \forall d \in [0, k] \colon |N_d(\mathfrak{X}, \mathbf{s})| = \sum_{I \subseteq [k], |I| = d} \prod_{i \in I} (\mathbf{s}_i - \mathbf{a}_i) \cdot \prod_{i \in [k] \setminus I} \mathbf{a}_i.$$

Proof. Fix an arbitrary vector attack $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$. Remember that $\mathfrak{X} = \mathfrak{X}_1 \times \ldots \times \mathfrak{X}_k$. For $i \in [k]$, there are $\mathbf{a}_i$ elements from $[\mathbf{s}_i]$ that are in $\mathfrak{X}_i$ and $(\mathbf{s}_i - \mathbf{a}_i)$ elements that are not. Each vector $\mathbf{v} \in \mathbb{V}(\mathbf{s})$ with *minimum* Hamming Distance d to $\mathfrak{X}$ has a unique corresponding combination $I \subseteq [k]$ with $|I| = d$, such that $\forall j \in I \colon \mathbf{v}_j \notin \mathfrak{X}_j$ and $\forall j \in [k] \setminus I \colon \mathbf{v}_j \in \mathfrak{X}_j$. For a fixed combination I, there are $\prod_{i \in I} (\mathbf{s}_i - \mathbf{a}_i) \cdot \prod_{i \in [k] \setminus I} \mathbf{a}_i$ such vectors and summing up over all possible $I \subseteq [k]$ of size d results in the size of $\mathfrak{X}$'s d-neighborhood. □

Claim 5.3.16
Let $\mathbf{a}$ be an attack distribution for a head distribution $\mathbf{s}$. Then it holds that

$$\forall \mathbf{v} \in \mathbb{V}(\mathbf{s}) \colon |\{\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s}) \mid \mathbf{v} \in N_d(\mathfrak{X}, \mathbf{s})\}| = \sum_{\substack{I \subseteq [k], |I| = k - d \\ \forall i \in I \colon \mathbf{a}_i \neq 0}} \prod_{i \in I} \binom{\mathbf{s}_i - 1}{\mathbf{a}_i - 1} \prod_{i \in [k] \setminus I} \binom{\mathbf{s}_i - 1}{\mathbf{a}_i} \tag{5.29}$$

Proof. Fix an arbitrary $\mathbf{v} \in \mathbb{V}(\mathbf{s})$. For each $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$ with $\mathbf{v} \in N_d(\mathfrak{X}, \mathbf{s})$, there is a unique set $I \subseteq [k]$ with $|I| = k - d$ such that $\forall i \in I \colon \mathbf{v}_i \in \mathfrak{X}_i$ and $\forall i \in [k] \setminus I \colon \mathbf{v}_i \notin \mathfrak{X}_i$. In particular, this means that $\forall i \in I \colon \mathbf{a}_i > 0$, since $\mathbf{v}_i > 0$ and $\mathfrak{X}_i = \{0\}$ if $\mathbf{a}_i = 0$.

Now fix such an index set I. For stripes $i \in I$, there are $\binom{\mathbf{s}_i - 1}{\mathbf{a}_i - 1}$ possibilities to choose $\mathfrak{X}_i \subseteq [\mathbf{s}_i]$ with $|\mathfrak{X}_i| = \mathbf{a}_i$ and $\mathbf{v}_i \in \mathfrak{X}_i$. For the stripes $i \in [k] \setminus I$, there are $\binom{\mathbf{s}_i - 1}{\mathbf{a}_i}$ possible ways to choose $\mathfrak{X}_i$ with $|\mathfrak{X}_i| = \mathbf{a}_i$ and $\mathfrak{X}_i \subseteq [\mathbf{s}_i] \setminus \mathbf{v}_i$. Since $\mathfrak{X} = \mathfrak{X}_1 \times \ldots \times \mathfrak{X}_k$, each different choice leads to a different vector attack $\mathfrak{X}$. Equation (5.29) is obtained by summing up over all possible sets I. □

Claim 5.3.17
Let $\mathbf{s}^a$ and $\mathbf{s}^b$ be head distributions, such that $\exists i, j \in [k]\colon\ (\mathbf{s}^b_i \leq \mathbf{s}^b_j) \wedge (\mathbf{s}^a_i = \mathbf{s}^b_i - 1) \wedge (\mathbf{s}^a_j = \mathbf{s}^b_j + 1) \wedge (\forall q \in [k] \setminus \{i, j\}\colon \mathbf{s}^a_q = \mathbf{s}^b_q)$. The following propositions are true:

1. Let $\mathbf{a}$ be an attack distribution for $\mathbf{s}^a$ with $\mathbf{a}_i \geq \mathbf{a}_j$. Then for every $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^a)$ and $\mathfrak{Y} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^b)$, it holds that

$$\forall q \in [0, k]\colon \sum_{r=0}^{q} \frac{|N_r(\mathfrak{X}, \mathbf{s}^a)|}{|\mathbb{V}(\mathbf{s}^a)|} \geq \sum_{r=0}^{q} \frac{|N_r(\mathfrak{Y}, \mathbf{s}^b)|}{|\mathbb{V}(\mathbf{s}^b)|}.$$

2. Let $\mathbf{a}^a$ and $\mathbf{a}^b$ be attack distributions for $\mathbf{s}^a$ and $\mathbf{s}^b$, respectively. In addition, they satisfy that for $c \in \{a, b\}\colon \mathbf{a}^c_i = \mathbf{s}^c_i$, $\forall p \in [k] \setminus \{i, j\}\colon \mathbf{a}^a_p = \mathbf{a}^b_p$, and

$$\mathbf{a}^a_j = \begin{cases} 0 & , \text{ if } \mathbf{a}^b_j = 0 \\ \mathbf{a}^b_j + 1 & , \text{ otherwise.} \end{cases}$$

   For every $\mathfrak{X} \in \mathbb{X}(\mathbf{a}^a, \mathbf{s}^a)$ and every $\mathfrak{Y} \in \mathbb{X}(\mathbf{a}^b, \mathbf{s}^b)$, it holds that

$$\forall q \in [0, k]\colon \sum_{r=0}^{q} \frac{|N_r(\mathfrak{X}, \mathbf{s}^a)|}{|\mathbb{V}(\mathbf{s}^a)|} \geq \sum_{r=0}^{q} \frac{|N_r(\mathfrak{Y}, \mathbf{s}^b)|}{|\mathbb{V}(\mathbf{s}^b)|}.$$

Figure 5.8 illustrates the relations of head and attack distributions in Claim 5.3.17. Furthermore, Example 5.3.18 shows an instance of a described situation.

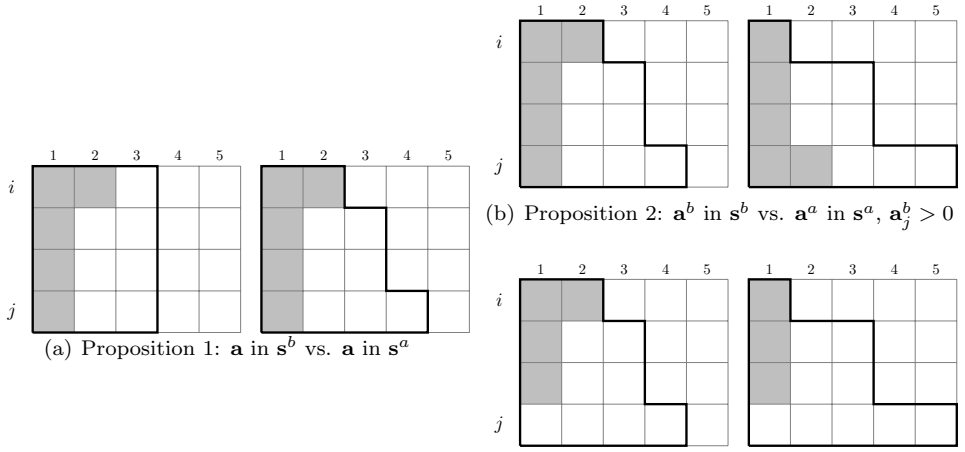
Example 5.3.18
For $C = 3$ and $k = 3$, a situation as specified in Proposition 1 of Claim 5.3.17 could have head distributions $\mathbf{s}^a = (2, 3, 4)$ and $\mathbf{s}^b = (3, 3, 3)$. Furthermore, assume an attack distribution $\mathbf{a} = (2, 0, 2)$.

Amongst others, we then have $\mathfrak{X} = \{1, 2\} \times \{0\} \times \{2, 4\} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^a)$ and $\mathfrak{Y} = \{1, 2\} \times \{0\} \times \{2, 3\} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^b)$. The neighborhoods up to distance 1 of these vector attacks are:

$$\begin{aligned} N_0(\mathfrak{X}, \mathbf{s}^a) &= N_0(\mathfrak{Y}, \mathbf{s}^b) = \emptyset \\ N_1(\mathfrak{X}, \mathbf{s}^a) &= \{1, 2\} \times [3] \times \{2, 4\} \\ N_1(\mathfrak{Y}, \mathbf{s}^b) &= \{1, 2\} \times [3] \times \{2, 3\} \end{aligned}$$

The vectors with distance up to 1 from $\mathfrak{X}$ have a higher share in $\mathbb{V}(\mathbf{s}^a)$, than the 'up-to-1 neighborhood' of $\mathfrak{Y}$ in $\mathbb{V}(\mathbf{s}^b)$:

$$\sum_{r=0}^{1} \frac{|N_r(\mathfrak{X}, \mathbf{s}^a)|}{|\mathbb{V}(\mathbf{s}^a)|} = \frac{0 + 12}{2 \cdot 3 \cdot 4} = \frac{1}{2} \geq \frac{4}{9} = \frac{0 + 12}{3 \cdot 3 \cdot 3} = \sum_{r=0}^{1} \frac{|N_r(\mathfrak{Y}, \mathbf{s}^b)|}{|\mathbb{V}(\mathbf{s}^b)|}$$



(a) Proposition 1: $\mathbf{a}$ in $\mathbf{s}^b$ vs. $\mathbf{a}$ in $\mathbf{s}^a$

(b) Proposition 2: $\mathbf{a}^b$ in $\mathbf{s}^b$ vs. $\mathbf{a}^a$ in $\mathbf{s}^a$, $\mathbf{a}^b_j > 0$

(c) Proposition 2: $\mathbf{a}^b$ in $\mathbf{s}^b$ vs. $\mathbf{a}^a$ in $\mathbf{s}^a$, $\mathbf{a}^b_j = 0$

Figure 5.8.: Schematic representation of situations as in Propositions 1 and 2 of Claim 5.3.17. Tupel values visualized by horizontal extent. Head distributions framed by thick line, attack distributions filled gray. In both cases the b-vectors are on the *left*!

Proof of Claim 5.3.17:

- Proposition 1: At first, note that $|\mathbb{V}(\mathbf{s}^a)| < |\mathbb{V}(\mathbf{s}^b)|$ since $\mathbf{s}^b_i \leq \mathbf{s}^b_j$ and

$$|\mathbb{V}(\mathbf{s}^a)| = \prod_{q\in[k]} \mathbf{s}^a_q = (\mathbf{s}^b_i - 1)(\mathbf{s}^b_j + 1) \prod_{q\in[k]\setminus\{i,j\}} \mathbf{s}^a_q \tag{5.30}$$

$$= \frac{(\mathbf{s}^b_i - 1)(\mathbf{s}^b_j + 1)}{\mathbf{s}^b_i \mathbf{s}^b_j} \mathbf{s}^b_i \mathbf{s}^b_j \prod_{q\in[k]\setminus\{i,j\}} \mathbf{s}^b_q \tag{5.31}$$

$$= \frac{\mathbf{s}^b_i \mathbf{s}^b_j - \mathbf{s}^b_j + \mathbf{s}^b_i - 1}{\mathbf{s}^b_i \mathbf{s}^b_j} |\mathbb{V}(\mathbf{s}^b)|. \tag{5.32}$$

  Next, for $c \in \{a, b\}$ and $\mathfrak{Z} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^c)$, we partition $\mathbb{V}(\mathbf{s}^c)$ based on the deviation from $\mathfrak{Z}$ in the positions i and j:

$$\mathbb{D}^c_{\emptyset}(\mathfrak{Z}) := \{\mathbf{v} \in \mathbb{V}(\mathbf{s}^c) \mid \mathbf{v}_i \in \mathfrak{Z}_i \wedge \mathbf{v}_j \in \mathfrak{Z}_j\} \tag{5.33}$$

$$\mathbb{D}^c_{\{i\}}(\mathfrak{Z}) := \{\mathbf{v} \in \mathbb{V}(\mathbf{s}^c) \mid \mathbf{v}_i \notin \mathfrak{Z}_i \wedge \mathbf{v}_j \in \mathfrak{Z}_j\} \tag{5.34}$$

$$\mathbb{D}^c_{\{j\}}(\mathfrak{Z}) := \{\mathbf{v} \in \mathbb{V}(\mathbf{s}^c) \mid \mathbf{v}_i \in \mathfrak{Z}_i \wedge \mathbf{v}_j \notin \mathfrak{Z}_j\} \tag{5.35}$$

$$\mathbb{D}^c_{\{i,j\}}(\mathfrak{Z}) := \{\mathbf{v} \in \mathbb{V}(\mathbf{s}^c) \mid \mathbf{v}_i \notin \mathfrak{Z}_i \wedge \mathbf{v}_j \notin \mathfrak{Z}_j\} \tag{5.36}$$

  Since the definition of the $\mathbb{D}$-sets implies a minimum distance from $\mathfrak{Z}$, they do

not intersect certain neighborhoods of $\mathfrak{Z}$:

$$N_0(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\{i\}}(\mathfrak{Z}) = \emptyset \tag{5.37}$$

$$N_0(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\{j\}}(\mathfrak{Z}) = \emptyset \tag{5.38}$$

$$(N_0(\mathfrak{Z}, \mathbf{s}^c) \cup N_1(\mathfrak{Z}, \mathbf{s}^c)) \cap \mathbb{D}^c_{\{i,j\}}(\mathfrak{Z}) = \emptyset \tag{5.39}$$

By assumption, we have $\forall q \in I \setminus \{i,j\} \colon \mathbf{s}^a_q = \mathbf{s}^b_q$. To shorten notation, let us introduce the following term for $d \in [0, k-2]$:

$$T(d) := \sum_{\substack{I \subseteq [k] \setminus \{i,j\} \\ |I| = d}} \prod_{q \in I} (\mathbf{s}^a_q - \mathbf{a}_q) \prod_{q \in [k] \setminus \{i,j\} \setminus I} \mathbf{a}_q. \tag{5.40}$$

For $c \in \{a, b\}$ and $\mathfrak{Z} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^c)$, it gives the number of vectors in $\mathbb{V}(\mathbf{s}^c)$ that have a fixed value tupel in positions i and j, and that deviate from $\mathfrak{Z}$ in exactly d other positions. Since $\mathbf{s}^a$ and $\mathbf{s}^b$ deviate only in the positions i and j, this value is the same for $\mathfrak{Z} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^a)$ and $\mathfrak{Z} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^b)$.

For every $d \in [0, k]$, it holds that

$$|N_d(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_\emptyset(\mathfrak{X})| = \mathbf{a}_i \mathbf{a}_j \cdot T(d) = |N_d(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_\emptyset(\mathfrak{Y})|. \tag{5.41}$$

Hence, the absolute number of vectors deviating from $\mathfrak{X}$, resp. $\mathfrak{Y}$, *neither* in position i *nor* j is equal in $N_d(\mathfrak{X}, \mathbf{s}^a)$ and $N_d(\mathfrak{Y}, \mathbf{s}^b)$. Furthermore, it holds that

$$\begin{aligned}
&|N_d(\mathfrak{X}, \mathbf{s}^a) \cap (\mathbb{D}^a_{\{i\}}(\mathfrak{X}) \cup \mathbb{D}^a_{\{j\}}(\mathfrak{X}))| \\
&= ((\mathbf{s}^a_i - \mathbf{a}_i)\mathbf{a}_j + (\mathbf{s}^a_j - \mathbf{a}_j)\mathbf{a}_i) \cdot T(d-1) && (5.42)\\
&= ((\mathbf{s}^b_i - 1 - \mathbf{a}_i)\mathbf{a}_j + (\mathbf{s}^b_j + 1 - \mathbf{a}_j)\mathbf{a}_i) \cdot T(d-1) && (5.43)\\
&= ((\mathbf{s}^b_i - \mathbf{a}_i)\mathbf{a}_j + (\mathbf{s}^b_j - \mathbf{a}_j)\mathbf{a}_i + (\mathbf{a}_i - \mathbf{a}_j)) \cdot T(d-1) && (5.44)\\
&= |N_d(\mathfrak{Y}, \mathbf{s}^b) \cap (\mathbb{D}^b_{\{i\}}(\mathfrak{Y}) \cup \mathbb{D}^b_{\{j\}}(\mathfrak{Y}))| + (\mathbf{a}_i - \mathbf{a}_j) \cdot T(d-1). && (5.45)
\end{aligned}$$

Since $\mathbf{a}_i \geq \mathbf{a}_j$, the absolute number of vectors in $N_d(\mathfrak{X}, \mathbf{s}^a)$ that deviate from $\mathfrak{X}$ in *either* position i *or* j, cannot be smaller than the number of such vectors in $N_d(\mathfrak{Y}, \mathbf{s}^b)$.

In contrast, we have

$$\begin{aligned}
&|N_d(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_{\{i,j\}}(\mathfrak{X})| \\
&= (\mathbf{s}^a_i - \mathbf{a}_i)(\mathbf{s}^a_j - \mathbf{a}_j) \cdot T(d-2) && (5.46)\\
&= (\mathbf{s}^b_i - 1 - \mathbf{a}_i)(\mathbf{s}^b_j + 1 - \mathbf{a}_j) \cdot T(d-2) && (5.47)\\
&= \Big((\mathbf{s}^b_i - \mathbf{a}_i)(\mathbf{s}^b_j - \mathbf{a}_j) + \mathbf{s}^b_i - \mathbf{s}^b_j + \mathbf{a}_j - \mathbf{a}_i - 1\Big) \cdot T(d-2) && (5.48)\\
&= |N_d(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_{\{i,j\}}(\mathfrak{Y})| - ((\mathbf{s}^b_j - \mathbf{s}^b_i) + (\mathbf{a}_i - \mathbf{a}_j) + 1) \cdot T(d-2). && (5.49)
\end{aligned}$$

Since $\mathbf{a}_j \leq \mathbf{a}_i < \mathbf{s}_i^b \leq \mathbf{s}_j^b$, this demonstrates that the absolute number of vectors in $N_d(\mathfrak{X}, \mathbf{s}^a)$ that deviate in both position i *and* j from $\mathfrak{X}$ is smaller than the number of such vectors in $N_d(\mathfrak{Y}, \mathbf{s}^b)$.

However, we are interested in the relative shares of neighborhoods in $\mathbb{V}(\mathbf{s}^a)$ and $\mathbb{V}(\mathbf{s}^b)$. Due to the Equations (5.37)–(5.39), for $c \in \{a, b\}$, $\mathfrak{Z} \in \mathbb{X}(\mathbf{a}, \mathbf{s}^c)$, and $q \in [0, k]$, we can make the following transformations:

$$\begin{aligned}
&\sum_{r=0}^{q} \frac{|N_r(\mathfrak{Z}, \mathbf{s}^c)|}{|\mathbb{V}(\mathbf{s}^c)|} \\
= &\sum_{r=0}^{q} \frac{|N_r(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\emptyset}(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|} + \sum_{r=0}^{q} \frac{|N_r(\mathfrak{Z}, \mathbf{s}^c) \cap (\mathbb{D}^c_{\{i\}}(\mathfrak{Z}) \cup \mathbb{D}^c_{\{j\}}(\mathfrak{Z}))|}{|\mathbb{V}(\mathbf{s}^c)|} \\
&+ \sum_{r=0}^{q} \frac{|N_r(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\{i,j\}}(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|} \\
= &\frac{|N_q(\mathfrak{Z}, \mathbf{s}^c) \cap (\mathbb{D}^c_{\emptyset}(\mathfrak{Z}) \cup \mathbb{D}^c_{\{i\}}(\mathfrak{Z}) \cup \mathbb{D}^c_{\{j\}}(\mathfrak{Z}))|}{|\mathbb{V}(\mathbf{s}^c)|} + \frac{|N_{q-1}(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\emptyset}(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|} \\
&+ \sum_{r=2}^{q} \frac{|N_{r-2}(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\emptyset}(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|} + \sum_{r=2}^{q} \frac{|N_{r-1}(\mathfrak{Z}, \mathbf{s}^c) \cap (\mathbb{D}^c_{\{i\}}(\mathfrak{Z}) \cup \mathbb{D}^c_{\{j\}}(\mathfrak{Z}))|}{|\mathbb{V}(\mathbf{s}^c)|} \\
&+ \sum_{r=2}^{q} \frac{|N_r(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\{i,j\}}(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|}
\end{aligned} \tag{5.50}$$

Due to $|\mathbb{V}(\mathbf{s}^a)| < |\mathbb{V}(\mathbf{s}^b)|$ and the Equations (5.41) and (5.45), we know that

$$\begin{aligned}
&\frac{|N_q(\mathfrak{X}, \mathbf{s}^a) \cap (\mathbb{D}^a_{\emptyset}(\mathfrak{X}) \cup \mathbb{D}^a_{\{i\}}(\mathfrak{X}) \cup \mathbb{D}^a_{\{j\}}(\mathfrak{X}))|}{|\mathbb{V}(\mathbf{s}^a)|} + \frac{|N_{q-1}(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_{\emptyset}(\mathfrak{X})|}{|\mathbb{V}(\mathbf{s}^a)|} \\
> &\frac{|N_q(\mathfrak{Y}, \mathbf{s}^b) \cap (\mathbb{D}^b_{\emptyset}(\mathfrak{Y}) \cup \mathbb{D}^b_{\{i\}}(\mathfrak{Y}) \cup \mathbb{D}^b_{\{j\}}(\mathfrak{Y}))|}{|\mathbb{V}(\mathbf{s}^b)|} + \frac{|N_{q-1}(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_{\emptyset}(\mathfrak{Y})|}{|\mathbb{V}(\mathbf{s}^b)|}.
\end{aligned} \tag{5.51}$$

Furthermore, for every $d \in [2, k]$, we can show the following equality:

$$\begin{aligned}
&\frac{|N_d(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_{\{i,j\}}(\mathfrak{X})|}{|\mathbb{V}(\mathbf{s}^a)|} + \frac{|N_{d-1}(\mathfrak{X}, \mathbf{s}^a) \cap (\mathbb{D}^a_{\{i\}}(\mathfrak{X}) \cup \mathbb{D}^a_{\{j\}}(\mathfrak{X}))|}{|\mathbb{V}(\mathbf{s}^a)|} \\
&+ \frac{|N_{d-2}(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_{\emptyset}(\mathfrak{X})|}{|\mathbb{V}(\mathbf{s}^a)|} \\
= &\frac{|N_d(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_{\{i,j\}}(\mathfrak{Y})|}{|\mathbb{V}(\mathbf{s}^a)|} + \frac{|N_{d-1}(\mathfrak{Y}, \mathbf{s}^b) \cap (\mathbb{D}^b_{\{i\}}(\mathfrak{Y}) \cup \mathbb{D}^b_{\{j\}}(\mathfrak{Y}))|}{|\mathbb{V}(\mathbf{s}^a)|} \\
&+ \frac{|N_{d-2}(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_{\emptyset}(\mathfrak{Y})|}{|\mathbb{V}(\mathbf{s}^b)|}
\end{aligned} \tag{5.52}$$

Note that $|\mathbb{V}(\mathbf{s}^b)| > |\mathbb{V}(\mathbf{s}^a)| > 0$ is true. If $T(d-2) = 0$, then both sides are zero (cmp. Equations (5.41),(5.44), and (5.48)). Otherwise, dividing both sides by

$$\frac{T(d-2)}{|\mathbb{V}(\mathbf{s}^b)|} \tag{5.53}$$

and using Equations (5.32), (5.41), (5.44), and (5.48) leaves

$$\begin{aligned}\frac{\mathbf{s}_i^b\mathbf{s}_j^b}{(\mathbf{s}_i^b-1)(\mathbf{s}_j^b+1)}\cdot\Big(&\big((\mathbf{s}_i^b-\mathbf{a}_i)(\mathbf{s}_j^b-\mathbf{a}_j)+\mathbf{s}_i^b-\mathbf{s}_j^b+\mathbf{a}_j-\mathbf{a}_i-1\big)\\ &+\big((\mathbf{s}_i^b-\mathbf{a}_i)\mathbf{a}_j+(\mathbf{s}_j^b-\mathbf{a}_j)\mathbf{a}_i)+\mathbf{a}_i-\mathbf{a}_j\big)+(\mathbf{a}_i\mathbf{a}_j)\Big)\\ =\ &\Big((\mathbf{s}_i^b-\mathbf{a}_i)(\mathbf{s}_j^b-\mathbf{a}_j)+(\mathbf{s}_i^b-\mathbf{a}_i)\mathbf{a}_j+(\mathbf{s}_j^b-\mathbf{a}_j)\mathbf{a}_i)+(\mathbf{a}_i\mathbf{a}_j)\Big),\end{aligned} \tag{5.54}$$

reducing to

$$\left(\frac{\mathbf{s}_i^b\mathbf{s}_j^b}{(\mathbf{s}_i^b-1)(\mathbf{s}_j^b+1)}\right)\cdot\big(\mathbf{s}_i^b\mathbf{s}_j^b\big)+\frac{\mathbf{s}_i^b\mathbf{s}_j^b(\mathbf{s}_i^b-\mathbf{s}_j^b-1)}{(\mathbf{s}_i^b-1)(\mathbf{s}_j^b+1)}=\mathbf{s}_i^b\mathbf{s}_j^b \tag{5.55}$$

$$\Leftrightarrow\ \left(\frac{(\mathbf{s}_i^b-1)(\mathbf{s}_j^b+1)}{(\mathbf{s}_i^b-1)(\mathbf{s}_j^b+1)}\right)\mathbf{s}_i^b\mathbf{s}_j^b=\mathbf{s}_i^b\mathbf{s}_j^b\quad\Leftrightarrow\quad\mathbf{s}_i^b\mathbf{s}_j^b=\mathbf{s}_i^b\mathbf{s}_j^b. \tag{5.56}$$

Hence, Equation (5.52) is true. Together with Equation (5.50) and Inequality (5.51), it leads to

$$\forall q\in[0,k]\colon\ \sum_{r=0}^{q}\frac{|N_r(\mathfrak{X},\mathbf{s}^a)|}{|\mathbb{V}(\mathbf{s}^a)|}\geq\sum_{r=0}^{q}\frac{|N_r(\mathfrak{Y},\mathbf{s}^b)|}{|\mathbb{V}(\mathbf{s}^b)|}. \tag{5.57}$$

- Proposition 2: The proof is similar to that of Proposition 1. Since $\mathbf{a}_i^a=\mathbf{s}_i^a$, there are no vectors in $\mathbb{V}(\mathbf{s}^a)$ that can deviate from $\mathfrak{X}$ in position i. The same applies to $\mathfrak{Y}$, since $\mathbf{a}_i^b=\mathbf{s}_i^b$. Hence, for arbitrary $c\in\{a,b\}$ and $\mathfrak{Z}\in\mathbb{X}(\mathbf{a}^c,\mathbf{s}^c)$, we know that

$$\mathbb{D}_{\{i\}}(\mathfrak{Z})=\emptyset \tag{5.58}$$

$$\mathbb{D}_{\{i,j\}}(\mathfrak{Z})=\emptyset. \tag{5.59}$$

Consequently, $\mathbb{V}(\mathbf{s}^c)$ is bipartitioned by $\mathbb{D}_{\emptyset}(\mathfrak{Z})$ and $\mathbb{D}_{\{j\}}(\mathfrak{Z})$.

The Equations (5.37)–(5.39) still apply and for $q\in[0,k]$ we can simplify Equation (5.50) to

$$\begin{aligned}&\sum_{r=0}^{q}\frac{|N_r(\mathfrak{Z},\mathbf{s}^c)|}{|\mathbb{V}(\mathbf{s}^c)|}\\ =&\frac{|N_q(\mathfrak{Z},\mathbf{s}^c)\cap\mathbb{D}_{\emptyset}^c(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|}+\sum_{r=1}^{q}\frac{|N_{r-1}(\mathfrak{Z},\mathbf{s}^c)\cap\mathbb{D}_{\emptyset}^c(\mathfrak{Z})|+|N_r(\mathfrak{Z})\cap\mathbb{D}_{\{j\}}^c(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|}.\end{aligned} \tag{5.60}$$

For $c \in \{a, b\}$ and $\mathfrak{Z} \in \mathbb{X}(\mathbf{a}^c, \mathbf{s}^c)$, we can write

$$\frac{|N_d(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_\emptyset(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|} = \frac{\mathbf{a}^c_j}{\mathbf{s}^c_j} \cdot \frac{\mathbf{a}^c_i}{\mathbf{s}^c_i} \cdot \frac{\sum\limits_{\substack{I \subseteq [k] \setminus \{i,j\} \\ |I| = d}} \prod\limits_{p \in I} (\mathbf{s}^c_p - \mathbf{a}^c_p) \prod\limits_{p \in [k] \setminus \{i,j\} \setminus I} \mathbf{a}^c_p}{\prod\limits_{p \in [k] \setminus \{i,j\}} \mathbf{s}^c_p} \tag{5.61}$$

and

$$\frac{|N_d(\mathfrak{Z}, \mathbf{s}^c) \cap \mathbb{D}^c_{\{j\}}(\mathfrak{Z})|}{|\mathbb{V}(\mathbf{s}^c)|} = \frac{(\mathbf{s}^c_j - \mathbf{a}^c_j)}{\mathbf{s}^c_j} \cdot \frac{\mathbf{a}^c_i}{\mathbf{s}^c_i} \cdot \frac{\sum\limits_{\substack{I \subseteq [k] \setminus \{i,j\} \\ |I| = d-1}} \prod\limits_{p \in I} (\mathbf{s}^c_p - \mathbf{a}^c_p) \prod\limits_{p \in [k] \setminus \{i,j\} \setminus I} \mathbf{a}^c_p}{\prod\limits_{p \in [k] \setminus \{i,j\}} \mathbf{s}^c_p}. \tag{5.62}$$

Due to our assumption that $\mathbf{a}^a_i = \mathbf{s}^a_i$, $\mathbf{a}^b_i = \mathbf{s}^b_i$, $\mathbf{a}^a_j = \mathbf{a}^b_j + \min(\mathbf{a}^b_j, 1)$, $\mathbf{s}^a_j = \mathbf{s}^b_j + 1$, $\mathbf{a}^a_j \leq \mathbf{s}^a_j$, and $\forall p \in [k] \setminus \{i, j\} \colon \mathbf{a}^a_p = \mathbf{a}^b_p \wedge \mathbf{s}^a_p = \mathbf{s}^a_p$ hold, for every $d \in [0, k]$, it follows from Equation (5.61) that

$$\frac{|N_d(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_\emptyset(\mathfrak{X})|}{|\mathbb{V}(\mathbf{s}^a)|} \geq \frac{|N_d(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_\emptyset(\mathfrak{Y})|}{|\mathbb{V}(\mathbf{s}^b)|}. \tag{5.63}$$

Furthermore, for $d \in [k]$ the Equations (5.61) and (5.62) lead to

$$\begin{aligned} &\frac{|N_{d-1}(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_\emptyset(\mathfrak{X})| + |N_d(\mathfrak{X}, \mathbf{s}^a) \cap \mathbb{D}^a_{\{j\}}(\mathfrak{X})|}{|\mathbb{V}(\mathbf{s}^a)|} \\ =&\frac{\sum_{\substack{I \subseteq [k] \setminus \{i,j\} \\ |I| = d-1}} \prod_{p \in I} (\mathbf{s}^a_p - \mathbf{a}^a_p) \prod_{p \in [k] \setminus \{i,j\} \setminus I} \mathbf{a}^a_p}{\prod_{p \in [k] \setminus \{i,j\}} \mathbf{s}^a_p} \\ =&\frac{|N_{d-1}(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_\emptyset(\mathfrak{Y})| + |N_d(\mathfrak{Y}, \mathbf{s}^b) \cap \mathbb{D}^b_{\{j\}}(\mathfrak{Y})|}{|\mathbb{V}(\mathbf{s}^b)|}. \end{aligned} \tag{5.64}$$

Hence, using Equations (5.60), (5.63), and (5.64), we obtain

$$\forall q \in [0, k] \colon \sum_{r=0}^{q} \frac{|N_r(\mathfrak{X}, \mathbf{s}^a)|}{|\mathbb{V}(\mathbf{s}^a)|} \geq \sum_{r=0}^{q} \frac{|N_r(\mathfrak{Y}, \mathbf{s}^b)|}{|\mathbb{V}(\mathbf{s}^b)|}. \tag{5.65}$$

□

Now let $\mathbf{s}$ be a head distribution and $\mathbf{a}$ be an attack distribution for $\mathbf{s}$. Furthermore, let $\mathcal{C} \in \mathbb{T}(n, C, k)$ have $\forall i \in [k] \colon |H^{\mathcal{C}}_i| = \mathbf{s}_i$ and let $X \subseteq H^{\mathcal{C}}$ be an arbitrary head attack with $\forall i \in [k] \colon |X \cap H^{\mathcal{C}}_i| = \mathbf{a}_i$.

The forward-damage $\mathrm{bf}^{\mathcal{C}}(X, z)$ for all $z \in [k]$ equals the total number of occurences

of the vectors $\bigcup_{d=0}^{k-z} N_d(\sigma(X), \mathbf{s})$ as rows in $M^{\mathcal{C}}$ (cmp. Equation (5.22)):

$$\begin{aligned}\mathrm{bf}^{\mathcal{C}}(X, z) &= |\{v \in V \mid \exists \mathbf{x} \in \sigma(X) \colon \mathrm{d}(M^{\mathcal{C}}[v], \mathbf{x}) \leq k - z\}| \\ &= \left|\left\{ v \in V \middle| M^{\mathcal{C}}[v] \in \bigcup_{d=0}^{k-z} N_d(\sigma(X), \mathbf{s}) \right\}\right|. \end{aligned} \tag{5.66}$$

Over all $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$, let us calculate the *average number* of occurences of the vectors $\bigcup_{d=0}^{k-z} N_d(\mathfrak{X}, \mathbf{s})$ as rows in $M^{\mathcal{C}}$. By Claim 5.3.15, the cardinality $\left|\bigcup_{d=0}^{k-z} N_d(\mathfrak{X}, \mathbf{s})\right|$ is equal for all $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$. Due to Claim 5.3.16, each $\mathbf{v} \in \mathbb{V}(\mathbf{s})$ is in minimum Hamming Distance up to $k - z$ with the same number of vector attacks $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$. Hence, the same applies to each of the n rows of $M^{\mathcal{C}}$. Using an arbitrary $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$, we can write

$$\begin{aligned}\sum_{v \in V} \sum_{\mathfrak{Y} \in \mathbb{X}(\mathbf{a},\mathbf{s})} \left|\bigcup_{d=0}^{k-z} N_d(\mathfrak{Y}, \mathbf{s}) \cap \{M^{\mathcal{C}}[v]\}\right| &= \sum_{v \in V} \sum_{\substack{\mathbf{v} \in \mathbb{V}(\mathbf{s}) \\ \mathbf{v} = M^{\mathcal{C}}[v]}} \sum_{\mathfrak{Y} \in \mathbb{X}(\mathbf{a},\mathbf{s})} \left|\bigcup_{d=0}^{k-z} N_d(\mathfrak{Y}, \mathbf{s}) \cap \{\mathbf{v}\}\right| \\ &= \sum_{v \in V} \sum_{\substack{\mathbf{v} \in \mathbb{V}(\mathbf{s}) \\ \mathbf{v} = M^{\mathcal{C}}[v]}} \frac{1}{\mathbb{V}(\mathbf{s})} \sum_{\mathfrak{Y} \in \mathbb{X}(\mathbf{a},\mathbf{s})} \left|\bigcup_{d=0}^{k-z} N_d(\mathfrak{Y}, \mathbf{s})\right| \\ &= \sum_{v \in V} \sum_{\substack{\mathbf{v} \in \mathbb{V}(\mathbf{s}) \\ \mathbf{v} = M^{\mathcal{C}}[v]}} \frac{|\mathbb{X}(\mathbf{a}, \mathbf{s})| \cdot \left|\bigcup_{d=0}^{k-z} N_d(\mathfrak{X}, \mathbf{s})\right|}{\mathbb{V}(\mathbf{s})} \\ &= n \cdot \frac{|\mathbb{X}(\mathbf{a}, \mathbf{s})| \cdot |\bigcup_{d=0}^{k-z} N_d(\mathfrak{X}, \mathbf{s})|}{\mathbb{V}(\mathbf{s})}. \end{aligned} \tag{5.67}$$

Dividing by $|\mathbb{X}(\mathbf{a}, \mathbf{s})|$, we obtain the *average number of occurences of the vectors* $\bigcup_{d=0}^{k-z} N_d(\mathfrak{X}, \mathbf{s})$ *as rows in* $M^{\mathcal{C}}$ *over all* $\mathfrak{X} \in \mathbb{X}(\mathbf{a}, \mathbf{s})$:

$$\overline{\mathrm{bf}}^{\mathcal{C}}(\mathfrak{X}, z) := \sum_{d=0}^{k-z} |N_d(\mathfrak{X}, \mathbf{s})| \cdot \frac{n}{|\mathbb{V}(\mathbf{s})|}. \tag{5.68}$$

This function plays a crucial role in the remaining part of this proof. In particular, it provides a lower bound on the maximum LoSS-damage of certain attacks on $\mathcal{C}$. Given any attack distribution $\mathbf{a}$ for $\mathbf{s}$, let $\mathfrak{Y} \in \operatorname{arg\,max}_{\mathfrak{Z} \in \mathbb{X}(\mathbf{a},\mathbf{s})} \sum_{v \in V} \left|\bigcup_{d=0}^{k-z} N_d(\mathfrak{Z}, \mathbf{s}) \cap \{M^{\mathcal{C}}[v]\}\right|$ and define

$$Y_{\mathfrak{Y}} := \bigcup_{i \in [k]} \{\sigma_i^{-1}(\mathbf{y}_i) \mid \mathbf{y} \in \mathfrak{Y} \wedge \mathbf{y}_i > 0\}. \tag{5.69}$$

It holds that $|Y_{\mathfrak{Y}}| \leq \sum_{i=1}^{k} \mathbf{a}_i$. Furthermore, we have $\mathfrak{Y}_i \subseteq \{\sigma_i(h) \mid h \in Y_{\mathfrak{Y}} \cap H_i^{\mathcal{C}}\}$, for

all $i \in [k]$. This leads to

$$\forall z \in [k] \colon \mathrm{bf}^{\mathcal{C}}(Y_{\mathfrak{Y}}, z) \geq \sum_{v \in V} \left| \bigcup_{d=0}^{k-z} N_d(\mathfrak{Y}, \mathbf{s}) \cap \{M^{\mathcal{C}}[v]\} \right| \geq \overline{\mathrm{bf}}^{\mathcal{C}}(\mathfrak{Y}, z). \tag{5.70}$$

In the following, we assume that $\mathcal{T} \in \mathbb{T}(n, C, k)$ is a topology with the properties given in Lemma 5.3.2 and with $M^{\mathcal{T}}$ being an $\mathrm{OA}(n, k, C, t)$.

Claim 5.3.19
For every attack $X \subseteq H^{\mathcal{T}}$ with $X \in \chi(\mathcal{T}, t)$, it holds that $\mathrm{bf}^{\mathcal{T}}(X, z) = \overline{\mathrm{bf}}^{\mathcal{T}}(\sigma(X), z)$.

Proof. The head distribution for $\mathcal{T}$ is $\mathbf{s}^{\mathcal{T}} = (C, \ldots, C)$ and there is an attack distribution $\mathbf{a}$ with $\forall i \in [k] \colon |X \cap H_i^{\mathcal{T}}| = \mathbf{a}_i$. Since $X \in \chi(\mathcal{T}, t)$ and since $\mathcal{T}$ has the one-stripe-only property, $\mathbf{a}$ has at most t non-zero values. In particular, there is $I \subseteq [k]$ with $|I| = t$ and $\{i \in [k] \mid X \cap H_i^{\mathcal{T}} \neq \emptyset\} \subseteq I$. Therefore, the Hamming Distance of every $\mathbf{v} \in \mathbb{V}(\mathbf{s}^{\mathcal{T}})$ to an $\mathbf{x} \in \sigma(X)$ will be at least $k - t$ and a node $v \in V$ is damaged only if $k - t \leq \min_{\mathbf{x} \in \sigma(X)} \mathrm{d}(M^{\mathcal{T}}[v], \mathbf{x}) \leq k - z$ (see Equation (5.22)).

Let $M_I^{\mathcal{T}}$ be the $n \times t$ submatrix of $M^{\mathcal{T}}$ consisting of the complete columns I and let $\mathbf{s}_I^{\mathcal{T}}$ be the restriction of $\mathbf{s}^{\mathcal{T}}$ to the positions I. Then, v is equivalently damaged if and only if $\min_{\mathbf{x} \in \sigma(X)_I} \mathrm{d}(M_I^{\mathcal{T}}[v], \mathbf{x}) \leq t - z$, where $\sigma(X)_I$ is the set of sub-vectors of $\sigma(X)$ in the columns I. Since $M^{\mathcal{T}}$ is an Orthogonal Array of strength t, each possible vector from $[C]^t$ occurs n/C^t times as a row vector in $M_I^{\mathcal{T}}$. This leads to

$$\mathrm{bf}^{\mathcal{T}}(X, z) = \sum_{d=0}^{t-z} |N_d(\sigma(X)_I, \mathbf{s}_I^{\mathcal{T}})| \cdot \frac{n}{C^t}. \tag{5.71}$$

For each vector $\mathbf{v} \in [C]^t$, there are C^{k-t} vectors in $\mathbb{V}(\mathbf{s}^{\mathcal{T}}) = [C]^k$ sharing $\mathbf{v}$ as a subvector in their positions I. Thus, it holds that $|N_{d+(k-t)}(\sigma(X), \mathbf{s}^{\mathcal{T}})| = |N_d(\sigma(X)_I, \mathbf{s}_I^{\mathcal{T}})| \cdot C^{k-t}$. Additionally applying Equation (5.68) we obtain

$$\mathrm{bf}^{\mathcal{T}}(X, z) = \sum_{d=0}^{t-z} |N_d(\sigma(X)_I, \mathbf{s}_I^{\mathcal{T}})| \cdot \frac{n}{C^t} = \sum_{d=0}^{k-z} |N_d(\sigma(X), \mathbf{s}^{\mathcal{T}})| \cdot \frac{n}{C^k} = \overline{\mathrm{bf}}^{\mathcal{T}}(\sigma(X), z). \tag{5.72}$$

□

Now let $\mathcal{C} \in \mathbb{T}(n, C, k)$ be a *witness against the t-forward-stability of* $\mathcal{T}$, i.e., for some attack size $x \in [Ck]$, some $z \in [k]$, and $I \subseteq [k]$ with $|I| = t$, it holds that

$$\max_{X \subseteq \bigcup_{i \in I} H_i^{\mathcal{T}}, |X| = x} \mathrm{bf}^{\mathcal{T}}(X, z) > \max_{Y \in \chi(\mathcal{C}, t), |Y| = x} \mathrm{bf}^{\mathcal{C}}(Y, z). \tag{5.73}$$

Here, the restriction to heads in $\mathcal{T}$ can be made due to Corollary 5.3.6.

For such witnesses, we can make a number of assumptions.

Claim 5.3.20
If there is a witness $\mathcal{C} \in \mathbb{T}(n, C, k)$ against the t-forward-stability of $\mathcal{T}$, then there is a witness $\mathcal{C}' \in \mathbb{T}(n, C, k)$ with depth 2 and $\sum_{i\in[k]} |H_i^{\mathcal{C}'}| = Ck$.

Proof. If $\mathrm{d}(\mathcal{C}) \neq 2$, there is $\mathcal{C}' \in \mathbb{T}(n, C, k)$ defined by

$$\forall i \in [k], \forall v \in V\colon \mathrm{parent}_i^{\mathcal{C}'}(v) := \begin{cases} \{s\} & , \text{ if } v \in H_i^{\mathcal{C}} \\ \mathrm{pred}_i^{\mathcal{C}}(v) \cap H_i^{\mathcal{C}} & , \text{ otherwise.} \end{cases} \tag{5.74}$$

The obtained topology $\mathcal{C}'$ satisfies $\forall v \in V, \forall i \in [k]\colon \mathrm{succ}_i^{\mathcal{C}'\rightarrow}(v) \subseteq \mathrm{succ}_i^{\mathcal{C}\rightarrow}(v)$, which leads to $\forall X \subseteq V\colon \mathrm{bf}^{\mathcal{C}'}(X, z) \leq \mathrm{bf}^{\mathcal{C}}(X, z)$. Since $\mathcal{C}$ is a witness, $\mathcal{C}'$ is a witness, too.

Now assume $\mathrm{d}(\mathcal{C}) = 2$ but $\sum_{i\in[k]} |H_i^{\mathcal{C}'}| < Ck$. Since $n \geq Ck$, there is $v \in V \setminus H^{\mathcal{C}}$. For a fixed $i \in [k]$, let $\{h\} = \mathrm{pred}_i^{\mathcal{C}}(v) \cap H_i^{\mathcal{C}}$. We create a topology $\mathcal{C}'$ that differs from $\mathcal{C}$ only in stripe i. In particular, $T_i' \in \mathcal{C}'$ is obtained from $T_i \in \mathcal{C}$ by making v an *additional*, childless head in T_i' and setting $\forall u \in \mathrm{child}_i^{\mathcal{C}}(v)\colon \mathrm{parent}_i^{\mathcal{C}'}(u) := \mathrm{parent}_i^{\mathcal{C}}(v)$. This leads to

$$\forall j \in [k], \forall u \in V \setminus \{v\}\colon \mathrm{succ}_j^{\mathcal{C}'\rightarrow}(v) \subseteq \mathrm{succ}_j^{\mathcal{C}\rightarrow}(h) \wedge \mathrm{succ}_j^{\mathcal{C}'\rightarrow}(u) \subseteq \mathrm{succ}_j^{\mathcal{C}\rightarrow}(u). \tag{5.75}$$

Study an arbitrary attack $X \in \chi(\mathcal{C}', t)$. If *either* $h \in X$ *or* $v \in X$, it holds that $\mathrm{bf}^{\mathcal{C}}(X \setminus \{v\} \cup \{h\}, z) \geq \mathrm{bf}^{\mathcal{C}'}(X, z)$. Otherwise, we have $\mathrm{bf}^{\mathcal{C}}(X, z) \geq \mathrm{bf}^{\mathcal{C}'}(X, z)$. In both cases, the attack on $\mathcal{C}$ is in $\chi(\mathcal{C}, t)$, since $X \in \chi(\mathcal{C}', t)$. Consequently, $\mathcal{C}'$ is again a witness, since $\mathcal{C}$ was a witness. Iterating this procedure leads to a topology $\mathcal{C}'$ with $\sum_{i=1}^{k} |H_i^{\mathcal{C}'}| = Ck$. □

Note that in a topology of depth 2, optimal attacks contain only heads, since all other nodes have empty forward successor sets.

In the following, we expect witness $\mathcal{C}$ to be a topology of depth 2 with $\sum_{i=1}^{k} |H_i^{\mathcal{C}}| = Ck$. Furthermore, there is a head distribution $\mathbf{s}^{\mathcal{C}}$ such that $\forall i \in [k]\colon |H_i^{\mathcal{C}}| = \mathbf{s}_i^{\mathcal{C}}$. W.l.o.g. we assume that $\forall i \in [k-1]\colon \mathbf{s}_i^{\mathcal{C}} \leq \mathbf{s}_{i+1}^{\mathcal{C}}$. This can be achieved by renaming the stripes of $\mathcal{C}$.

Claim 5.3.21
For every attack $X \in \chi(\mathcal{T}, t)$, there is an attack $Y \subseteq H^{\mathcal{C}}$ with $Y \in \chi(\mathcal{C}, t)$ and $|Y| \leq |X|$, such that $\mathrm{bf}^{\mathcal{C}}(Y, z) \geq \mathrm{bf}^{\mathcal{T}}(X, z)$.

Proof. Due to Corollary 5.3.6, it suffices to study attacks $X \subseteq H^{\mathcal{T}}$ only. Let $\mathbf{a}^{\mathcal{T}}$ be the attack distribution with $\forall i \in [k]\colon \mathbf{a}_i^{\mathcal{T}} = |X \cap H_i^{\mathcal{T}}|$. W.l.o.g., we can assume that $\forall i \in [k-1]\colon \mathbf{a}_i^{\mathcal{T}} \geq \mathbf{a}_{i+1}^{\mathcal{T}}$ (otherwise rename the stripes of $\mathcal{T}$). Since $\mathcal{T}$ has the one-stripe-only property, it holds that $\sum_{i=1}^{k} \mathbf{a}_i^{\mathcal{T}} = |X|$. Furthermore, $\mathbf{a}^{\mathcal{T}}$ has at most t non-zero entries because $X \in \chi(\mathcal{T}, t)$ additionally holds.

Now observe that $\mathbf{s}^{\mathcal{T}} = (C, \ldots, C)$ can be transformed into $\mathbf{s}^{\mathcal{C}}$ by iterating the following operation: given a head distribution $\mathbf{s}^b$, return an altered head distribution $\mathbf{s}^a$ with, for distinct $i, j \in [k]$, $\mathbf{s}_i^a := \mathbf{s}_i^b - 1$, $\mathbf{s}_j^a := \mathbf{s}_j^b + 1$ and $\forall q \in [k] \setminus \{i, j\}\colon \mathbf{s}_q^a := \mathbf{s}_q^b$.

By tracing the *shortest* sequence of such operations transforming $\mathbf{s}^{\mathcal{T}}$ into $\mathbf{s}^{\mathcal{C}}$, we can *simultaneously* transform $\mathbf{a}^{\mathcal{T}}$ into an attack distribution $\mathbf{a}^{\mathcal{C}}$ for $\mathbf{s}^{\mathcal{C}}$. Let $\mathbf{a}^b$ be the attack distribution *before* a modification of $\mathbf{s}^a$ in positions $i, j \in [k]$. Then we return an altered $\mathbf{a}^a$ with $\forall q \in [k] \setminus \{i,j\}\colon \mathbf{a}^a_q := \mathbf{a}^b_q$,

$$\mathbf{a}^a_i := \begin{cases} \mathbf{a}^b_i & , \text{ if } \mathbf{s}^a_i \geq \mathbf{a}^b_i \\ \mathbf{a}^b_i - 1 & , \text{ otherwise,} \end{cases} \quad \text{and} \quad \mathbf{a}^a_j := \begin{cases} \mathbf{a}^b_j & , \text{ if } \mathbf{s}^a_i \geq \mathbf{a}^b_i \\ \mathbf{a}^b_j + 1 & , \text{ if } \mathbf{s}^a_i < \mathbf{a}^b_i \wedge \mathbf{a}^b_j > 0 \\ 0 & , \text{ otherwise.} \end{cases} \tag{5.76}$$

Note that for each operation, the pairs $(\mathbf{a}^b, \mathbf{s}^b)$ and $(\mathbf{a}^a, \mathbf{s}^a)$ correspond to the situation in either Proposition 1 or 2 of Claim 5.3.17. Both were visualized in Figure 5.8. In particular, it holds that $\forall p \in [k]\colon \mathbf{a}^b_p = 0 \Leftrightarrow \mathbf{a}^a_p = 0$. Since the sequence of operations starts with $\mathbf{a}^{\mathcal{T}}$ and ends with $\mathbf{a}^{\mathcal{C}}$, we also obtain $\forall p \in [k]\colon \mathbf{a}^{\mathcal{T}}_p = 0 \Leftrightarrow \mathbf{a}^{\mathcal{C}}_p = 0$.

Now let $\mathfrak{Y} \in \arg\max_{\mathfrak{Z} \in \mathbb{X}(\mathbf{a}^{\mathcal{C}}, \mathbf{s}^{\mathcal{C}})} \sum_{v \in V} \left| \bigcup_{d=0}^{k-z} N_d(\mathfrak{Z}, \mathbf{s}^{\mathcal{C}}) \cap \{M^{\mathcal{C}}[v]\} \right|$ and let $Y_{\mathfrak{Y}}$ be defined as in Equation (5.69). It holds that $|Y_{\mathfrak{Y}}| \leq \sum_{i=1}^{k} \mathbf{a}^{\mathcal{C}}_i \leq \sum_{i=1}^{k} \mathbf{a}^{\mathcal{T}}_i = |X|$. Furthermore, each node in $Y_{\mathfrak{Y}}$ is head of $\mathcal{C}$ in at least one stripe $p \in [k]$ with $\mathbf{a}^{\mathcal{C}}_p \neq 0$. Since both $\mathbf{a}^{\mathcal{T}}$ and $\mathbf{a}^{\mathcal{C}}$ have at most t non-zero entries, it holds that $Y_{\mathfrak{Y}} \in \chi(\mathcal{C}, t)$.

The claim is proven by showing the following:

$$\mathrm{bf}^{\mathcal{C}}(Y_{\mathfrak{Y}}, z) \geq \overline{\mathrm{bf}}^{\mathcal{C}}(\mathfrak{Y}, z) = n \cdot \frac{\sum_{d=0}^{k-z} |N_d(\mathfrak{Y}, \mathbf{s}^{\mathcal{C}})|}{|\mathbb{V}(\mathbf{s}^{\mathcal{C}})|} \tag{5.77}$$

$$\geq n \cdot \frac{\sum_{d=0}^{k-z} |N_d(\sigma(X), \mathbf{s}^{\mathcal{T}})|}{|\mathbb{V}(\mathbf{s}^{\mathcal{T}})|} \tag{5.78}$$

$$= \overline{\mathrm{bf}}^{\mathcal{T}}(\sigma(X), z) = \mathrm{bf}^{\mathcal{T}}(X, z). \tag{5.79}$$

For this, we use Inequality (5.70) and Equation (5.68) in Line (5.77). Applying Propositions 1 and 2 of Claim 5.3.17 on $(\mathbf{a}^b, \mathbf{s}^b)$ and $(\mathbf{a}^a, \mathbf{s}^a)$ for each step of the above-mentioned sequence of operations, we obtain Line (5.78). Finally, Equation (5.68) and Claim 5.3.19 lead to the result in Line (5.79). □

Claim 5.3.21 contradicts the assumption that $\mathcal{C}$ was a witness against the t-forward-stability of $\mathcal{T}$. Therefore, no such witness can exist in $\mathbb{T}(n, C, k)$ and $\mathcal{T}$ must be t-forward-stable. □

In the following, we will show that if Orthogonal Arrays $\mathrm{OA}(n, k, C, t)$ exist, their use is necessary for the forward-stability of topologies $\mathcal{T} \in \mathbb{T}(n, C, k)$. This will follow from an important result about generalizations of Orthogonal Arrays.

Definition 5.3.22 ***Packing Array [BB12]***
For $n, k, C, t, \lambda \in \mathbb{N}$, an $n \times k$ matrix M with entries $m_{vi} \in [C]$ is called a *Packing Array* $\mathrm{PA}_\lambda(n, k, C, t)$, if in every $n \times t$ submatrix M' consisting of t complete columns of M, each $\mathbf{x} \in [C]^t$ appears *at most* λ *times* as a row.

A $\mathrm{PA}_\lambda(\lambda C^t, k, C, t)$ is an $\mathrm{OA}(\lambda C^t, k, C, t)$, since the maximum row frequency is limited to the average frequency over all possible combinations of t-vectors over $[C]$. Due to an argument very similar to that of Corollary 5.3.11, every $\mathrm{PA}_\lambda(n, k, C, t)$ is a $\mathrm{PA}_{\lambda C}(n, k, C, t-1)$. Regarding the minimum possible values of λ, we can show the following Lemma.

Lemma 5.3.23
For $n, k, C, t \in \mathbb{N}$ and $t < k$, let λ_t resp. λ_{t+1} be the smallest values λ such that a Packing Array $\mathrm{PA}_\lambda(n, k, C, t)$ resp. $\mathrm{PA}_\lambda(n, k, C, t+1)$ exists.
It holds that $\lambda_t \geq \lambda_{t+1} \geq \left\lceil \frac{\lambda_t}{C} \right\rceil$.

Proof. Let M_t be a $\mathrm{PA}_{\lambda_t}(n, k, C, t)$ and M_{t+1} be a $\mathrm{PA}_{\lambda_{t+1}}(n, k, C, t+1)$.

Assume that $\lambda_{t+1} < \left\lceil \frac{\lambda_t}{C} \right\rceil$. Since M_{t+1} is also a $\mathrm{PA}_{C\lambda_{t+1}}(n, k, C, t)$, it holds that $C\lambda_{t+1} \leq C \cdot (\left\lceil \frac{\lambda_t}{C} \right\rceil - 1) < \lambda_t$. This is a contradiction with the assumption that λ_t is minimal.

Now assume $\lambda_{t+1} > \lambda_t$, fix $t+1$ arbitrary columns of M_t and count the frequencies of their rows. The frequency of a row in any t of these $t+1$ columns is limited to λ_t. Additionally considering the $(t+1)$-th column, the maximum frequency of a row cannot be higher than λ_t, since in the worst case (the complete $(t+1)$-th row contains the same entry) it would be just the maximum frequency of a row in the t other columns. Thus, M_t is a $\mathrm{PA}_{\lambda_t}(n, k, C, t+1)$. However, this is a contradiction with the assumption that λ_{t+1} is minimal. □

This inequality is strict, since Orthogonal Arrays of strength $t+1$ achieve the lower bound and the upper bound is always hit for Orthogonal Arrays of index 1 and strength $t < k$. The following theorem shows that Packing Arrays of minimum λ are important for forward-stability.

Theorem 5.3.24
Let $\mathbb{M}(n, k, C, p)$ be the set of Packing Arrays $\mathrm{PA}_\lambda(n, k, C, p)$ with

$$\lambda = \min\{\lambda \in \mathbb{N} \mid \text{a } \mathrm{PA}_\lambda(n, k, C, p) \text{ exists}\}.$$

Every t-forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ has $M^{\mathcal{T}} \in \left(\bigcap_{p=1}^{t} \mathbb{M}(n, k, C, p)\right)$.

Proof. Assume $M^{\mathcal{T}}$ is not in $\bigcap_{p=1}^{t} \mathbb{M}(n, k, C, p)$. Then there is $p \leq t$, such that $M^{\mathcal{T}} \notin \mathbb{M}(n, k, C, p)$. Due to Corollary 5.3.1, the t-forward-stable topology $\mathcal{T}$ has to minimize the maximum forward-damage for attacks of cardinality p and $z = p$. We show that, under the above assumption, this is not the case.

$\mathcal{T}$ must have the properties listed in Lemma 5.3.2. Furthermore, let $\mathcal{C} \in \mathbb{T}(n, C, k)$ be a topology with the same properties and $M^{\mathcal{C}} \in \mathbb{M}(n, k, C, p)$.

For $z = p$, study the possible forward-damage of attacks of cardinality p. Such an attack may target heads from less than p different stripes, leading to forward-damage of 0. Alternatively, it can target one head from each stripe of a combination of p stripes. The maximum forward-damage of the latter attacks on $\mathcal{T}$ and $\mathcal{C}$ equals the maximum

n	k	C	t	Method	Remark
C^t	$C+1$	$C \geq t$	t	[Bus52]/Reed-Solomon	C is prime power
λC^t	$t+1$	C	t	Zero-Sum Arrays	$\lambda \in \mathbb{N}$
$p^{(t-1)m+q}$	$p^m + \left\lfloor \frac{m}{q} \right\rfloor$	p^q	t	[Bie96]	p prime, $\substack{2 \leq t \leq p^m \\ q \leq m}$
C^{k-m}	$\frac{C^m-1}{C-1}$	C	$C^{m-1}-1$	Rao-Hamming constr.	C is prime power
2^{3m-1}	2^m	2	7	Delsarte-Goethals codes	
2^{2m+1}	2^m+1	2	5	BCH codes	$m \geq 5$
C^4	C^2+1	C	3	[HSS99, pp. 101-102]	C is prime power
8λ	4λ	2	3	Hadamard Arrays	$\lambda \in \mathbb{N}$

Table 5.1.: An incomplete list of generic constructions for Orthogonal Arrays. Generally $m \in \mathbb{N}$. See [HSS99] for details.

row frequency in $M_I^{\mathcal{T}}$ resp. $M_{I'}^{\mathcal{C}}$ over all $I, I' \subseteq [k], |I| = |I'| = p$. An attack achieving this damage contains the heads corresponding to the entries of the most frequent row vector. Since $\mathcal{C} \in \mathbb{M}(n,k,C,p) \not\ni \mathcal{T}$, the damage value is smaller on $\mathcal{C}$ than on $\mathcal{T}$. Hence, $\mathcal{T}$ is not t-forward-stable. □

Note that we cannot generally assume $\mathbb{M}(n,k,C,t) = \bigcap_{p=1}^{t} \mathbb{M}(n,k,C,p)$. Following the notation of Lemma 5.3.23, we could have the case that $\lambda_{t-1} = qC + r$ for $q \in \mathbb{N}$ and $r \in [C-1]$. Then, we had $\lambda_t \geq (q+1)$. Hence, the matrices $\mathbb{M}(n,k,C,t)$ would all be $\mathrm{PA}_{(q+1)C}(n,k,C,t)$ but need not be $\mathrm{PA}_{\lambda_{t-1}}(n,k,C,t-1)$. Thus, they do not have to be part of $\mathbb{M}(n,k,C,t-1)$.

Theorem 5.3.24 has the following corollary as a special case.

Corollary 5.3.25
If an $\mathrm{OA}(n,k,C,t)$ exists, then for every t'-forward-stable $\mathcal{T} \in \mathbb{T}(n,C,k)$ with $t' \geq t$, $M^{\mathcal{T}}$ is an $\mathrm{OA}(n,k,C,t)$.

Proof. For an $\mathrm{OA}(n,k,C,t)$ M, the maximum row frequency of a vector $\mathbf{v} \in [C]^t$ in any t-column submatrix of M equals the average value $\frac{n}{C^t}$ of these frequencies. Thus, it cannot be decreased further. If $\mathrm{OA}(n,k,C,t)$ exist, these are exactly the matrices in $\mathbb{M}(n,k,C,t)$. Since each $\mathrm{OA}(n,k,C,t)$ is an $\mathrm{OA}(n,k,C,t-1)$, we especially have $\mathbb{M}(n,k,C,t) = \bigcap_{i=1}^{t} \mathbb{M}(n,C,k,i)$. □

Hence, when aiming to create a forward-stable distribution topology $\mathcal{T} \in \mathbb{T}(n,C,k)$, we have to find the maximum t such that an $\mathrm{OA}(n,k,C,t)$ exists. For this, we can rely on the existing theory on Orthogonal Arrays (e.g., [CD06, HSS99] and references therein), which offers both a high number of generic constructions based on Galois Fields and finite geometries, combinatorics, and error-correcting codes, together with lists of specific Orthogonal Arrays that do not follow one of the known constructions. Table 5.1 lists some of the generic constructions known so far. It is easy to see that they are only available for specific parameter combinations. At least, we can still use the concatenation property of Corollary 5.3.12 to form Orthogonal Arrays having more

rows. The general decision on the existence of Orthogonal Arrays for given parameters will be a topic of Section 5.3.4.

5.3.3. Connections with Error-Correcting Codes

Despite our progress on necessary and sufficient conditions for t-forward stability in Subsection 5.3.2, important questions are still unaddressed. In particular, we do not exactly know which topologies $\mathcal{T} \in \mathbb{T}(n, C, k)$ with matrices $M^{\mathcal{T}} \in \bigcap_{p=1}^{k} \mathbb{M}(n, k, C, p)$ are indeed forward-stable.

A possible approach to shed light on this question is to introduce the interpretation of $M^{\mathcal{T}}$ as a matrix listing an error-correcting code. This is the topic of this subsection. Along the way, we will give a review of necessary and connected coding-theoretical results and mention how to efficiently determine the strength of a given matrix. This is a premise for results in Subsection 5.3.4. Then, we identify Maximum Distance Separable (MDS) codes as a class of codes with very beneficial stability characteristics. We will show important properties of these codes and suggest studying their isometry classes, since they categorize behavior of topologies $\mathcal{T} \in \mathbb{T}(n, C, k)$ under attacks from $\mathcal{P}(V) \setminus \chi(\mathcal{T}, t)$ where t is the strength of $M^{\mathcal{T}}$ as an Orthogonal Array.

At first, we introduce codes.

Definition 5.3.26 ***Code*** M ***[Bie05, Definition 1.8]***
Let A be a finite set and let $k \in \mathbb{N}$. A *code of length k over alphabet A* is a set $M \subseteq A^k$. Each $\mathbf{v} \in M$ is called *codeword*.

Generally, we will not distinguish between a code and a matrix listing all codewords. Thus, we will denote both by M and write $\mathbf{v} \in M$ if $\mathbf{v}$ is a row of M. Figure 5.9 gives an example of a code.

A code over an alphabet of cardinality C is called *C-ary*. The *strength* of a code is equivalent to the strength of the matrix M in terms of Orthogonal Arrays. The *minimum distance* of a code is the minimum of pairwise Hamming Distances between distinct codewords.

Since we aim at finding forward-stable topologies, in the following we generally assume that the studied topologies $\mathcal{T} \in \mathbb{T}(n, C, k)$ have the properties given in Lemma 5.3.2. Now fix an arbitrary $\mathcal{T} \in \mathbb{T}(n, C, k)$ with these properties. If each $\mathbf{v} \in [C]^k$ appears *at most once* as a row of $M^{\mathcal{T}}$, the matrix $M^{\mathcal{T}}$ is code of length k over the alphabet $[C]$.

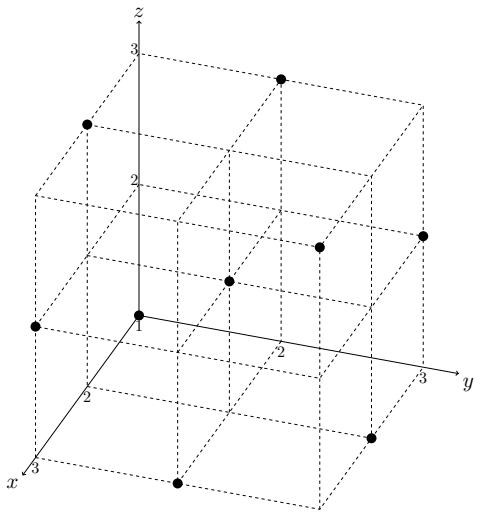
It is easy to see the following:

Corollary 5.3.27
For every forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, the matrix $M^{\mathcal{T}}$ is a code if $n \leq C^k$.

Proof. This follows from Theorem 5.3.24 and the fact that for $n \leq C^k$ the set $\mathbb{M}(n, C, k, k)$ is exactly the set of codes over the alphabet $[C]$ which have n codewords and length k. □

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 2 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \\ 3 & 2 & 1 \\ 3 & 3 & 3 \end{pmatrix}$$

(a) M as a matrix



(b) M in $[3]^3$

Figure 5.9.: A code M over alphabet [3] with 9 codewords of length 3. M is isometric to an MDS code.

A Technical Assumption The forward-damage $\mathrm{bf}^{\mathcal{T}}(X, z)$ of a head attack $X \subseteq H^{\mathcal{T}}$ is equal to the number of $M^{\mathcal{T}}$'s codewords in a Hamming Distance up to $k - z$ from $\sigma(X)$ (see Equation (5.22)).

If there is a stripe $i \in [k]$ such that $X \cap H_i^{\mathcal{T}} = \emptyset$, this will lead to a difference in vector position i between every $\mathbf{x} \in \sigma(X)$ and every $\mathbf{v} \in [C]^k$. It follows that

$$\mathrm{bf}^{\mathcal{T}}(X, z) = \mathrm{bf}^{\mathcal{T} \setminus \{T_i\}}(X, z), \tag{5.80}$$

since on topology $\mathcal{T} \setminus \{T_i\}$ the distance limit drops to $k - z - 1$ and, for every node $v \in V$, we have $\mathrm{d}(\sigma(X), M^{\mathcal{T} \setminus \{T_i\}}[v]) = \mathrm{d}(\sigma(X), M^{\mathcal{T}}[v]) - 1$.

Hence, an attack X with $\exists I \subset [k]$ such that $X \subseteq \bigcup_{i \in I} H_i^{\mathcal{T}}$ can be analyzed on a substitutional topology $\biguplus_{i \in I} \{T_i\}$ consisting only of the stripes I of $\mathcal{T}$. *We will therefore, from now on, assume, that every attack on a topology $\mathcal{T}$ targets at least one head of each stripe of $\mathcal{T}$.* This gives the technical advantage that $\sigma(X) \subseteq [C]^k$.

Minimum Distances In this setting, the stability properties of $M^{\mathcal{T}}$ are determined by the arrangement of its codewords in the metric space $([C]^k, \mathrm{d})$, where d is the Hamming distance. Each head attack $X \subseteq H^{\mathcal{T}}$ corresponds to a set of vectors $\sigma(X)$ from $[C]^k$ and for each vector $\mathbf{x} \in [C]^k$ there is a head attack X with $\sigma(X) = \{\mathbf{x}\}$. The forward-damage of an attack X equals the number of codewords within a Hamming distance up to $k - z$ from any of the vectors $\sigma(X)$. Consequently, the code $M^{\mathcal{T}}$ of a forward-stable topology $\mathcal{T}$ must have a high minimum distance d, since $\lceil \frac{d}{2} \rceil$ is the

minimum distance such that we can find an $\mathbf{x} \in [C]^k$ (i.e., a possible attack) having more than one of $M^{\mathcal{T}}$'s codewords (rows) in within this distance.

Code Isometries When studying $M^{\mathcal{T}}$ as a code, we should also consider code isometry. A mapping $\phi\colon A \to B$ between two metric spaces (A, d_1) and (B, d_2) is an *isometry*, if it is distance-preserving, i.e., $\forall a, b \in A\colon \mathrm{d}_1(a,b) = \mathrm{d}_2(\phi(a), \phi(b)))$. In the following, we will generally restrict to isometries regarding the Hamming distance metric. As a result of [CH96, Theorem 1], a mapping from $[C]^k$ to $[C]^k$ is an such an isometry if and only if it consists of a permutation of the vector positions (i.e., columns) and in each position an independent renaming of the code letters.

For two $n \times k$-matrices M_1, M_2 with entries from $[C]$, we will write $\phi(M_1) = M_2$ if $\forall v \in [n]\colon \phi(M_1[v]) = M_2[v]$. For the study of forward-stability of topologies $\mathcal{T} \in \mathbb{T}(n, C, k)$, isometries $\phi\colon [C]^k \to [C]^k$ are interesting not only when considering $M^{\mathcal{T}}$ to be a code.

Lemma 5.3.28 ***Isometries of*** $M^{\mathcal{T}}$
Let $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}(n, C, k)$ be distribution topologies with the properties listed in Lemma 5.3.2 and let ϕ be an isometry $\phi\colon [C]^k \to [C]^k$ with $\phi(M^{\mathcal{T}_1}) = M^{\mathcal{T}_2}$.

For every head attack $X \subseteq H^{\mathcal{T}_1}$ there is a head attack $Y \subseteq H^{\mathcal{T}_2}$ and a permutation π of $[k]$, such that it holds that

$$\forall i \in [k]\colon |X \cap H_i^{\mathcal{T}_1}| = |Y \cap H_{\pi(i)}^{\mathcal{T}_2}| \text{ and } \forall z \in [k]\colon \mathrm{bf}^{\mathcal{T}_1}(X, z) = \mathrm{bf}^{\mathcal{T}_2}(Y, z).$$

Proof. For $i \in [k]$, let $\sigma_i^{\mathcal{T}_1}$ and $\sigma_i^{\mathcal{T}_2}$ be the bijections used to construct $M^{\mathcal{T}_1}$ and $M^{\mathcal{T}_2}$. By [CH96], the isometry ϕ consists of a permutation π of the vector positions of each vector in $[C]^k$ and bijections $\phi_1, \ldots, \phi_k\colon [C] \to [C]$ such that ϕ_i renames the letters of position i *after* the permuation. We construct Y from X as

$$Y := \bigcup_{i \in [k]} \left\{ h \in H_{\pi(i)}^{\mathcal{T}_2} \,\middle|\, \exists x \in X \cap H_i^{\mathcal{T}_1}\colon \sigma_{\pi(i)}^{\mathcal{T}_2}(h) = \phi_{\pi(i)}(\sigma_i^{\mathcal{T}_1}(x)) \right\}. \tag{5.81}$$

This definition ensures that $\phi(\sigma^{\mathcal{T}_1}(X)) = \sigma^{\mathcal{T}_2}(Y)$ and $\forall i \in [k]\colon |X \cap H_i^{\mathcal{T}_1}| = |Y \cap H_{\pi(i)}^{\mathcal{T}_2}|$.

Since ϕ is an isometric, bijective mapping from $[C]^k$ to $[C]^k$, it holds that

$$\begin{aligned}
\mathrm{bf}^{\mathcal{T}_1}(X, z) &= |\{v \in V \mid \exists \mathbf{x} \in \sigma^{\mathcal{T}_1}(X)\colon \mathrm{d}(M^{\mathcal{T}_1}[v], \mathbf{x}) \leq k - z\}| && (5.82)\\
&= |\{v \in V \mid \exists \mathbf{x} \in \phi(\sigma^{\mathcal{T}_1}(X))\colon \mathrm{d}(\phi(M^{\mathcal{T}_1}[v]), \mathbf{x}) \leq k - z\}| && (5.83)\\
&= |\{v \in V \mid \exists \mathbf{x} \in \sigma^{\mathcal{T}_2}(Y)\colon \mathrm{d}(M^{\mathcal{T}_2}[v], \mathbf{x}) \leq k - z\}| = \mathrm{bf}^{\mathcal{T}_2}(Y, z). && (5.84)
\end{aligned}$$

□

Hence, a topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ is t-forward-stable if and only if there is a t-forward-stable topology $\mathcal{C} \in \mathbb{T}(n, C, k)$ and an isometry $\phi\colon [C]^k \to [C]^k$ such that $\phi(M^{\mathcal{C}}) = M^{\mathcal{T}}$. Consequently, when studying the forward-stability of topologies whose matrix $M^{\mathcal{T}}$ is a code, it suffices to study the equivalence classes of these codes under isometries from $[C]^k$ to $[C]^k$.

Linear Codes Codes with especially interesting properties are the linear codes. They are defined with the Galois field $\mathbb{F}_C$ as their alphabet. Although the codes $M^{\mathcal{T}}$ studied so far had the alphabet $[C]$, this alphabet change is only of a technical nature. Every numbering of the elements of $\mathbb{F}_C$ is a bijection between $\mathbb{F}_C$ and $[C]$. Using one of them to rename the code letters admits a bijective isometry mapping $\mathbb{F}_C^k$ to $[C]^k$ (and with it any code inside these spaces).

Definition 5.3.29 ***Linear Code [Bie05, Definition 3.6]***
Let $\mathbb{F}_C$ be the Galois field with C elements. A linear subspace $M \subseteq \mathbb{F}_C^k$ of vector space dimension t is called a C-ary linear code of *length* k and *dimension* t.

As a vector subspace of $\mathbb{F}_C^k$, each linear code M is characterized by a *base* consisting of t linearly independent vectors from M. The codewords of M are the linear combinations (with computations in $\mathbb{F}_C$) of these base vectors. Hence, M has C^t codewords.

One of the most important classes of linear codes are the (Extended) Reed-Solomon Codes [RS60, MS93]. They can be used in the construction of Orthogonal Arrays (cmp. Table 5.1) and are possible candidates for stream error correction (see Section 2.1.2).

Linear codes have the following property.

Corollary 5.3.30
For each codeword $\mathbf{v}$ of a code M, define $\mathbf{d}(\mathbf{v}) = (d_0, \ldots, d_k)$ with $d_i := |\{\mathbf{w} \in M \mid d(\mathbf{v}, \mathbf{w}) = i\}|$ as the *distance distribution* of $\mathbf{v}$ in M. If M is a linear code, then for every pair $\mathbf{v}, \mathbf{w} \in M$ it holds that $\mathbf{d}(\mathbf{v}) = \mathbf{d}(\mathbf{w})$.

Proof. Adding $\mathbf{w} - \mathbf{v}$ to every codeword of M is an isometry $\phi\colon M \to M$ (here, a special renaming of code letters in each position) and maps $\mathbf{v}$ to $\mathbf{w}$. □

Codes with the above property are called *distance-invariant*.

Dual Distance and the Strength of Matrices The *dual code* $M^\perp$ of a linear code M is another C-ary linear code of length k such that $\mathbf{u} \in M^\perp \Leftrightarrow \forall \mathbf{v} \in M\colon \mathbf{u} \cdot \mathbf{v}^T = 0$, where $\cdot$ is the scalar product. The *dual distance* $d^\perp$ *of* M is the minimum distance of $M^\perp$. The concept of dual distance is also defined for non-linear codes. However, in this case the definition is based on a transformation of the average distance distribution of the code (see [MS93] for more details).

For a code M, the following result of Delsarte provides a link between $d^\perp$ and the strength of M as an Orthogonal Array.

Lemma 5.3.31 ***Delsarte's Theorem [HSS99, Theorem 4.9]***
If M is a (not necessarily linear) C-ary code with n codewords of length k and dual distance $d^\perp$ then M is an $\mathrm{OA}(n, k, C, d^\perp - 1)$. If M is a code and an $\mathrm{OA}(n, k, C, t)$ then M has dual distance $d^\perp \geq t + 1$.

Albeit originally defined for codes, Delsarte's Theorem also applies if we generalize our notion of a code to allow multiple instances of a codeword (with the result that the

rows of any matrix over $[C]$ become a C-ary code). Therefore, it provides a method to determine the strength of a given $n \times k$ matrix M in time $O(n^2k)$: This can be done by counting the pairwise Hamming Distances between the rows of M, determining an average distance distribution, establishing the dual distance distribution as the so-called McWilliams transform of the average distance distribution and reading out $d^\perp$ from the result (see [HSS99] Chapters 4.1/4.4 for details).

MDS codes We have already seen that a forward-stable topology $\mathcal{T}$ with $n \leq C^k$ will maximize the minimum distance of the code $M^{\mathcal{T}}$. This distance is limited by the Singleton Bound. Codes satisfying it with equality are called MDS codes.

Lemma 5.3.32 ***Singleton Bound [Bie05, Theorem 4.1]***
If M is a C-ary code of length k, minimum distance d, and n codewords, then it satisfies $n \leq C^{k-d+1}$.

Definition 5.3.33 ***Maximum Distance Separable Code [MS93]***
A C-ary code of length k, having C^t codewords, and minimum distance d is called *maximum distance separable* (or MDS) if $d = k - t + 1$.

Again, the (Extended) Reed-Solomon codes give examples of MDS codes. Furthermore, each $\mathrm{OA}(C^t, k, C, t)$ M is an MDS code, since it is a code and every two codewords have at least distance $k - t + 1$. Otherwise, they would coincide in at least t positions which is impossible in an Orthogonal Array of index 1 and strength t.

There is a long-standing conjecture about the maximum length of MDS codes. Under a different terminology, its disputed part was first stated in [Seg55].

Conjecture 5.3.34 ***The MDS Conjecture [Rot06]***
Let $k(C, t)$ be the maximum length of a C-ary MDS code having C^t codewords. If C is a prime power, it holds that

$$k(C,t) = \begin{cases} \infty & \text{, if } t = 1 \\ C+1 & \text{, if } t \in [C-2] \setminus \{1,3\} \\ C+1 & \text{, if } t \in \{3, C-1\} \text{ and } C \text{ is odd,} \\ C+2 & \text{, if } t \in \{3, C-1\} \text{ and } C \text{ is even,} \\ t+1 & \text{, if } t \geq C. \end{cases}$$

MDS codes not only maximize their minimum distance but also the dual distance. The following lemma was first shown by [Sil60] and is here formulated using our terminology.

Lemma 5.3.35 ***[Sil60, Lemma 2]***
A C-ary MDS code with C^t codewords has strength t.

Hence, a C-ary MDS-code with length k and C^t codewords is an $\mathrm{OA}(C^t, k, C, t)$ *and vice versa.* It is a code with the highest possible minimum distance and the highest possible strength (with respect to its parameters).

Corollary 5.3.36
If a C-ary MDS code with C^t codewords and length k exists, then for every t'-forward-stable topology $\mathcal{T} \in \mathbb{T}(C^t, C, k)$ with $t' \geq t$, the code $M^{\mathcal{T}}$ is an MDS code.

Proof. This is a direct consequence of Corollary 5.3.25 and the equivalence of MDS codes and Orthogonal Arrays, both of strength t, in the given setting. □

Furthermore, *every* MDS code M with C^t codewords and length k is also contained in the sets $\mathbb{M}(C^t, k, C, t')$ (see Theorem 5.3.24) for $k \geq t' > t$. This follows since M is an $\mathrm{OA}(C^t, k, C, t)$ of index 1 and since, due to Lemma 5.3.23, the maximum frequency of every vector $\mathbf{v} \in [C]^{t'}$ in any t' columns of M also meets the minimum possible value $\lceil C^t / C^{t'} \rceil = 1$. Hence, all such MDS codes satisfy the requirement that Theorem 5.3.24 states for the matrix $M^{\mathcal{T}}$ of a forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$.

Consequently, if C-ary MDS codes with C^t codewords and length k exist, their equivalence classes under isometries from $[C]^k$ to $[C]^k$ define the complete set of candidate matrices $M^{\mathcal{T}}$ for forward-stable topologies $\mathcal{T} \in \mathbb{T}(C^t, C, k)$.

In Appendix B, we prove that every C-ary MDS code M with C^t codewords and length k is distance-invariant and has the same distance distribution $\mathbf{d}_{C,k,t}(\cdot)$. Based on this, we can show that that all topologies $\mathcal{T} \in \mathbb{T}(n, C, k)$ with the properties given in Lemma 5.3.2 and a matrix $M^{\mathcal{T}}$ listing an MDS code suffer equal forward-damage from vector attacks coinciding with single codewords of $M^{\mathcal{T}}$.

Corollary 5.3.37
Let $\mathcal{T} \in \mathbb{T}(n, C, k)$ be a topology with the properties given in Lemma 5.3.2 and with $M^{\mathcal{T}}$ being an MDS code. Furthermore, fix an arbitrary $\mathbf{v} \in M^{\mathcal{T}}$. For each attack $X \subseteq H^{\mathcal{T}}$ with $\forall i \in [k] : |X \cap H_i^{\mathcal{T}}| = 1$ and $\sigma(X) \subseteq M^{\mathcal{T}}$, it holds that

$$\forall z \in [k] : \mathrm{bf}^{\mathcal{T}}(X, z) = \sum_{i=0}^{k-z} \mathbf{d}_{C,k,t}(\mathbf{v})_i.$$

Proof. The vector attack $\sigma(X)$ contains a single codeword from $M^{\mathcal{T}}$. The damage of X follows from Equation (5.22) and Lemma B.0.4. □

However, for two arbitrary MDS codes M_1 and M_2 with alphabet $[C]$, length k, and C^t codewords, it is not guaranteed that there is an isometry $\phi : [C]^k \to [C]^k$ with $\phi(M_1) = M_2$. Hence, there could be other attacks for which topologies $\mathcal{T}$ and $\mathcal{C}$ with $M^{\mathcal{T}} = M_1$ and $M^{\mathcal{C}} = M_2$ suffer different values of forward-damage.

On Isometry Classes of MDS Codes We have seen that if $n = C^t$ and if MDS codes of length k and n codewords exist, a forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ must have a matrix $M^{\mathcal{T}}$ being such a code. However, these codes may behave differently when considering their t'-forward-stability for $t' > t$. We have also seen that we only have to consider one representative from each equivalence class of MDS codes under isometries from $[C]^k$ to $[C]^k$.

It would be beneficial to show, that for certain parameters of MDS codes there is but one such isometry class (and thus every such code has equal stability properties). However, no non-trivial parameters with this property are known to the author.

There *are* considerable research results (e.g., [OD85, BBF$^+$06]) on the isometry classes of MDS codes. Unfortunately, these concentrate on studying equivalence classes of *linear* MDS codes under *(semi-)linear* isometries, i.e., isometries limited to a column permutation and a columnwise multiplication with elements of $\mathbb{F}_C^*$ (for semilinear isometries this is followed by a field automorphism of $\mathbb{F}_C$ in each column). Every such isometry $\phi' \colon M_1 \to M_2$ between C-ary linear MDS codes M_1 and M_2 of same length k and dimension t can be extended to an isometry $\phi \colon \mathbb{F}_C^k \to \mathbb{F}_C^k$ of the whole space $\mathbb{F}_C^k$ with $\phi(M_1) = M_2$ (exactly the isometries we are interested in). As a result of this research, representatives from the equivalence classes of MDS codes under (semi-)linear isometries can be described by *systematic generator matrices* [OD85, BBF$^+$06] having a certain normalized form.

Among other things, this implies that all *linear* MDS codes of length k and dimension $k-1$ are in the same equivalence class under linear isometries. Thus, the same applies to equivalence classes under isometries from $\mathbb{F}_C^k$ to $\mathbb{F}_C^k$. However, this still does not include non-linear MDS codes with these parameters.

New results in this field could promote our knowledge on forward-stable topologies.

5.3.4. (In-)Existence of Orthogonal Arrays and Complexity of Finding Forward-Stable Topologies

As a result of Section 5.3.2, we know that to construct a forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, we have to determine the maximum value t such that an $\mathrm{OA}(n, k, C, t)$ exists. Let $\hat{\mathrm{t}}(n, k, C)$ be this value and let $\hat{\mathrm{k}}(n, C, t)$ be the maximum value of k such that an $\mathrm{OA}(n, k, C, t)$ exists.

We have shown in Corollary 5.3.11 that an $\mathrm{OA}(n, k, C, t)$ is also an $\mathrm{OA}(n, k, C, t-1)$. There is no $\mathrm{OA}(n, k, C, t)$ with $k < t$ and we can use an $\mathrm{OA}(n, k, C, t)$ to obtain an $\mathrm{OA}(n, k', C, t)$ with $k \geq k' \geq t$ by dropping $k - k'$ columns. Consequently, the existence of an Orthogonal Array $\mathrm{OA}(n, k, C, t)$ could be determined by using $\hat{\mathrm{k}}(n, C, t)$ or $\hat{\mathrm{t}}(n, k, C)$, given the parameters n, C, t or n, C, k, respectively.

With this observation, both functions are related as follows:

$$\hat{\mathrm{t}}(n, k, C) = \max\{t \in [0, \min(k, \lfloor \log_C n \rfloor)] \mid \hat{\mathrm{k}}(n, C, t) \geq k\} \tag{5.85}$$

$$\hat{\mathrm{k}}(n, C, t) = \max\{k \in \mathbb{N} \mid k \geq t \wedge \hat{\mathrm{t}}(n, k, C) \geq t\} \tag{5.86}$$

Hence, if one of them could be computed efficiently, so could the other (e.g., using

(one-sided) binary search).

Similar to the above, we can furthermore introduce $\hat{\mathrm{n}}(n, k, C, t)$ as the minimum value $n^* \geq n$ such that an $\mathrm{OA}(n^*, k, C, t)$ exists and define $\hat{\mathrm{n}}(k, C, t) := \hat{\mathrm{n}}(0, k, C, t)$. Both functions are defined for each combination of $C \in \mathbb{N}, k \in \mathbb{N}$, and $t \in [0, k]$. The function $\hat{\mathrm{n}}(k, C, t)$ can be determined from $\hat{\mathrm{k}}(n, C, t)$:

$$\hat{\mathrm{n}}(k, C, t) = \min\{n \in [C^k] \mid \hat{\mathrm{k}}(n, C, t) \geq k\}. \tag{5.87}$$

There are several constraints known for the existence of Orthogonal Arrays. By definition, we must have $n = \lambda C^t$ for some $\lambda \in \mathbb{N}$. Furthermore, there are the following general bounds, which can all be used to obtain upper bounds on $\hat{\mathrm{k}}(n, C, t)$ and $\hat{\mathrm{n}}(k, C, t)$.

Lemma 5.3.38 ***Rao bound [CD06]***
An $\mathrm{OA}(\lambda C^t, k, C, t)$ exists only if

$$n \geq \begin{cases} 1 + \sum_{i=1}^{t/2} \binom{k}{i}(C-1)^i & \text{, if } t \text{ even} \\ 1 + \sum_{i=1}^{(t-1)/2} \binom{k}{i}(C-1)^i + \binom{k-1}{(t-1)/2}(C-1)^{(t+1)/2} & \text{, if } t \text{ odd.} \end{cases}$$

Lemma 5.3.39 ***Bush bound [CD06]***
An $\mathrm{OA}(C^t, k, C, t)$ with $t > 1$ exists only if

$$k \leq \begin{cases} C + t - 1 & \text{, if } C \text{ even and } t \leq C \\ C + t - 2 & \text{, if } C \text{ odd and } 3 \leq t \leq C \\ t + 1 & \text{, if } t \geq C. \end{cases}$$

Additionally, there is the Bose-Bush bound [CD06], which is specific for $t \in \{2, 3\}$, and several bounds based on solutions of Linear Programs first introduced by Delsarte (see [HSS99, Chapter 4.5] for details).

However, none of the identified bounds is known to be exact! Given n, C and t, they allow us to obtain upper bounds on $\hat{\mathrm{k}}(n, C, t)$, but in a very large number of cases there is a considerable gap between all these bounds and the largest number k, such that an $\mathrm{OA}(n, k, C, t)$ is actually *known* to exist. The same applies to $\hat{\mathrm{n}}(k, C, t)$.

This is a fundamental problem in Design Theory and in [HSS99, p.32] the investigation of better bounds for $\hat{\mathrm{k}}(n, C, t)$ and $\hat{\mathrm{n}}(k, C, t)$ is stated as Research Problem 2.32. Although, since then, better bounds have been identified for special cases, the general problem is yet unresolved.

When restricting to Orthogonal Arrays that are codes, the identification of exact bounds would either prove or disprove the *MDS Conjecture* (see Conjecture 5.3.34) in Coding Theory. It claims to specify maximum lengths for MDS codes with given alphabet size and dimension. This conjecture stands unproven since its formulation in 1955 [Seg55] (there stated in the terminology of projective geometries).

Given the status of the *existence* problem of Orthogonal Arrays, it is not surprising that also the general *construction* problem of Orthogonal Array is still widely unsolved.

Research Problem 12.11 of [HSS99] asks for algorithms that can actually construct an Orthogonal Array with given parameters, provided that such an array exists. Research Problem 12.12 asks for time complexity bounds of such an algorithm. Given that the inputs are n, k, C, t and that the result is an $n \times k$ matrix, it must have at least pseudopolynomial runtime. All algorithmic approaches known to the author resort to meta-heuristics [Gon07] and local search schemes [MFL00, Xu02, NL08].

As we have seen in Equation (5.85), these circumstances also hinder us to generally determine $\hat{\mathrm{t}}(n, k, C)$ efficiently. Consequently, we arrive at a point where both existence and construction of forward-stable topologies, for a high number of parameter combinations, depend on long-standing open questions from Design and Coding Theory. This dependency also holds from a reversed point of view. We show that solving the Restricted Forward-Stable Topology Formation Problem is at least as hard as the general construction problem of Orthogonal Arrays. Furthermore, the existence of a pseudopolynomial algorithm solving it, would imply pseudopolynomial algorithms to compute $\hat{\mathrm{t}}(n, k, C)$ and $\hat{\mathrm{k}}(n, C, t)$:

Theorem 5.3.40
Let $\mathcal{O}$ be an oracle for the Restricted Forward-Stable Topology Formation Problem.

- If one exists, an $\mathrm{OA}(n, k, C, t)$ can be constructed by one call to $\mathcal{O}$ plus $O(nk)$-time post-processing.
- The function $\hat{\mathrm{t}}(n, k, C)$ can be evaluated by $\lceil \log(k) \rceil$ calls to $\mathcal{O}$ plus $O(n^2 k)$-time post-processing.
- The function $\hat{\mathrm{k}}(n, C, t)$ can be evaluated by $\lceil \log(n) \rceil \cdot \lceil \log(k) \rceil$ calls to $\mathcal{O}$ plus $O(n^2 k)$-time post-processing per call.

Proof. By Theorem 5.3.14 and Corollary 5.3.25, there is a t-forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ if an $\mathrm{OA}(n, k, C, t)$ exists. In this case, $M^{\mathcal{T}}$ must be an $\mathrm{OA}(n, k, C, t)$. Given n, C, k, t as input, such a $\mathcal{T}$ is obtained by one call to $\mathcal{O}$. The information necessary to return the $n \times k$ matrix $M^{\mathcal{T}}$ can be gathered by a traversal of all stripe trees. This needs time $O(nk)$.

Applying binary search, we need $\lceil \log(k) \rceil$ oracle calls to determine the maximum $t' \in [k]$ such that a t'-forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ exists. By Corollary 5.3.25, $M^{\mathcal{T}}$ must be an $\mathrm{OA}(n, k, C, \hat{\mathrm{t}}(n, k, C))$. Due to Lemma 5.3.31 (and the algorithm stemming from it) we can compute the strength of $M^{\mathcal{T}}$ in time $O(n^2 k)$.

Using this procedure as a subroutine, we can compute $\hat{\mathrm{k}}(n, C, t)$ with Equation (5.86) and at most $\lceil \log(n) \rceil$ steps in a binary search. □

In this thesis, we will not be able to resolve the problem of computing $\hat{\mathrm{t}}(n, k, C)$ efficiently. Thus, in many cases it still remains unknown whether an $\mathrm{OA}(n, k, C, t)$ exists (without investing huge amounts of computing resources).

The Case of Packing Arrays In Theorem 5.3.24 we have seen, that every forward-stable topology must be a Packing Array minimizing its maximum row frequency.

Since Orthogonal Arrays are Packing Arrays, the limits shown above also apply to the existence problem for this generalization. Beyond that, there are scarcely any (in-)existence results for this general class of matrices. The only results known to the author restrict to the case $t = 2$ [SM02] and a table in [CD06] listing parameter combinations for which Packing Arrays exist.

5.4. Summary

In this chapter, we have made important steps towards the general understanding of optimally LoSS-stable and forward-stable distribution topologies. Since LoSS- and FEC-LoSS-damage of an attack do not differ on topologies of depth at most 2, the results are also relevant for the minimization of FEC-LoSS-damage.

The findings of this chapter were published in [Gra12]

In Section 5.1, we defined optimally LoSS-stable topologies and determined their basic properties. In the subsequent Section 5.2, we analyzed the LoSS-damage function. In particular, we demonstrated that LoSS-damage can be seen as a superimposition of two different types of damage: the *direct damage* and the *forward-damage*. For topologies with $n \gg Ck$ nodes, the latter dominates the LoSS-damage measure. Based on this observation, we defined forward-stable topologies, minimizing the maximum possible forward-damage for every choice of attack parameters. In the following Section 5.3, we focused on the study of these topologies.

In Subsection 5.3.1, we identified first necessary requirements and made sure that they do not impede the construction of optimally LoSS-stable topologies. The forward-stability of topologies satisfying these requirements is characterized by the forward successor sets of their heads. In Subsection 5.3.2, we introduced a matrix repesentation for these sets. We proved that, for forward-stable topologies, these matrices must be Orthogonal Arrays resp. Packing Arrays with a minimum maximum subrow frequency. Furthermore, we could show that Orthogonal Arrays of strength t guarantee t-forward-stable topologies.

For peer numbers $n \leq C^k$, Subsection 5.3.3 then interpreted these matrices as error-correcting codes. Thereby, we could transfer results from Coding Theory to our study of forward-stable distribution topologies. We saw that if there is a C-ary MDS code with C^t codewords and length k, every forward-stable topology $\mathcal{T} \in \mathbb{T}(C^t, C, k)$ must have a matrix $M^{\mathcal{T}}$ whose rows list such a code.

We pointed out that the isometry classes of the studied matrices (and codes) can be used to categorize their suitability to construct forward-stable topologies. In particular, for C-ary MDS codes of C^t codewords, codes from different isometry classes may lead to different values of maximum forward-damage from attacks on heads in more than t stripes. Therefore, we briefly reviewed the available research results on isometry classes of MDS codes. However, albeit there are a number of relevant results in the literature, we could not use them to find non-trivial parameters for which only a single such isometry class exists.

Since the matrix $M^{\mathcal{T}}$ of a forward-stable topology $\mathcal{T}$ must be an Orthogonal Array of maximum strength, Subsection 5.3.4 studied extremal parameter combinations for

Orthogonal Arrays. We saw that the problem of determining the existence of Orthogonal Arrays for arbitrary parameters is an unsolved problem in Design Theory. In particular, its special case – the MDS Conjecture – stimulates (not only) coding theorists since 1955. We proved that efficiently finding (t-)forward-stable topologies in arbitrary classes $\mathbb{T}(n, C, k)$ would also allow to efficiently solve the aforementioned problems. Due to this reason, such an efficient construction currently remains unknown.

Incompatibilities with Cluster Topologies and Rule-Based Topologies With the obtained results, we have to realize that the requirements for both the Cluster Topologies and the rule-based topologies of Chapter 4 can prevent the construction of forward-stable topologies.

In a Cluster Topology $\mathcal{T} \in \mathbb{T}(n, C, k)$, the matrix $M^{\mathcal{T}}$ will contain only C different types of row vectors. In particular, all $\lceil n/C \rceil$ resp. $\lfloor n/C \rfloor$ rows corresponding to nodes of the same cluster have the same entries, since all these nodes have the same preceding heads. Consequently, for a Cluster Topology with $k > 1$, the matrix $M^{\mathcal{T}}$ will not satisfy the requirements given in Theorem 5.3.24.

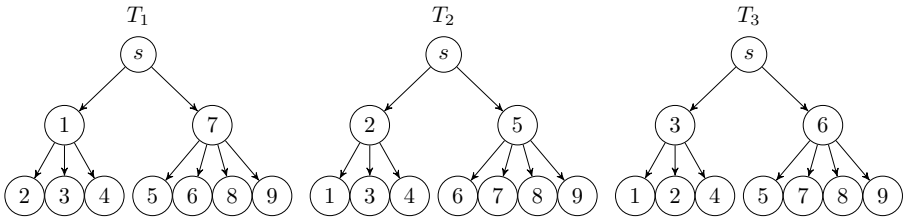
For rule-based topologies, the requirements of Head Rule 2 can lead to negative effects by creating highly dependent groups of heads. This phenomenon especially appears in topology classes $\mathbb{T}(n, C, k)$ with $n \mod C = 1$. Here, the head topology of a rule-based topology $\mathcal{T}$ will contain k heads that pairwise supply each other. These are the heads with $\delta_1^{C,k} = \lceil n/C \rceil + k - 1$ successors. They share a common row vector in $M^{\mathcal{T}}$. If $n < k \cdot C^k$, the frequency k of this row vector can be too high to form topologies meeting the demands given in Theorem 5.3.24.

However, in contrast to the Cluster Topologies, the deviation from optimal matrices $M^{\mathcal{T}}$ is limited. The maximum row frequency is increased by at most $k - 1$ and does not grow with n. The deviation occurs only for rows corresponding to head nodes and can be compensated if $n \geq k \cdot C^k$.

Figure 5.10 gives an example for the described incompatibilities. It shows representative topologies from $\mathbb{T}(9, 2, 3)$. In this class, all Cluster Topologies and rule-based topologies have a layout similar to the topologies $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. For $z = 2$ and $x = 2$, a worst-case attack on all shown topologies is given by $X = \{4, 6\}$. It leads to forward-damage of $\mathrm{bf}^{\mathcal{T}_1}(X, 2) = 5$, $\mathrm{bf}^{\mathcal{T}_2}(X, 2) = 4$, and $\mathrm{bf}^{\mathcal{T}_3}(X, 2) = 3$, respectively. Thus, neither $\mathcal{T}_1$ nor $\mathcal{T}_2$ can be forward-stable.

The Cluster Topologies also conflict with optimal LoSS-stability. This is caused by the fact that their massive deviations from optimal forward-damage cannot be compensated by direct damage. Again, Figure 5.10 provides an example. Considering LoSS-damage, for $z = 2$ and $x = 2$, a worst-case attack on both $\mathcal{T}_1$ and $\mathcal{T}_3$ is $X = \{4, 6\}$. It leads to LoSS-damage of $\mathrm{b}^{\mathcal{T}_1}(X, 2) = 5$ and $\mathrm{b}^{\mathcal{T}_3}(X, 2) = 4$. Thus, $\mathcal{T}_1$ cannot be optimally LoSS-stable in $\mathbb{T}(9, 2, 3)$.

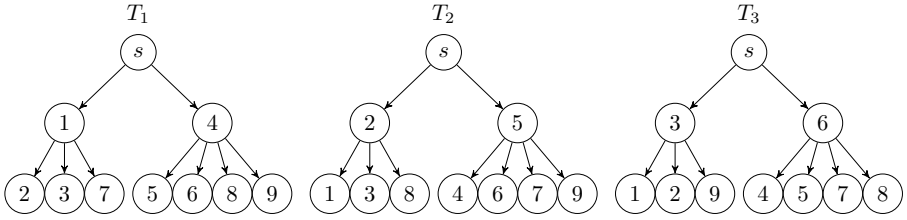
For rule-based topologies, the compatibility with LoSS-stability is unknown. In particular, it is possible that the limited deviations from optimal values of forward-damage can always be compensated by the influence of direct damage. Again, such an effect can be seen in Figure 5.10, where the maximum LoSS-damage for $x = 2$ and $z = 2$ is 4 on both $\mathcal{T}_2$ and $\mathcal{T}_3$. It is achieved by attack $X = \{4, 6\}$.



(a) Cluster Topology $\mathcal{T}_1$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

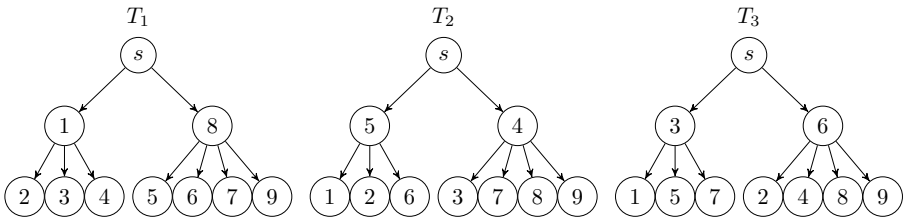
(b) ${M^{\mathcal{T}_1}}^T$ for $\sigma_1(1) = \sigma_2(2) = \sigma_3(3) = 1$ and $\sigma_1(4) = \sigma_2(5) = \sigma_3(6) = 2$.



(c) Rule-based topology $\mathcal{T}_2$

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & 1 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 1 \end{pmatrix}$$

(d) ${M^{\mathcal{T}_2}}^T$ for $\sigma_1(1) = \sigma_2(2) = \sigma_3(3) = 1$ and $\sigma_1(4) = \sigma_2(5) = \sigma_3(6) = 2$.



(e) Alternative topology $\mathcal{T}_3$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 2 \end{pmatrix}$$

(f) ${M^{\mathcal{T}_3}}^T$ for $\sigma_1(1) = \sigma_2(5) = \sigma_3(3) = 1$ and $\sigma_1(8) = \sigma_2(4) = \sigma_3(6) = 2$.

Figure 5.10.: Example that Cluster Topologies and rule-based topologies are not forward-stable in $\mathbb{T}(9, 2, 3)$. Note the high frequencies of particular row vectors in $M^{\mathcal{T}_1}$ and $M^{\mathcal{T}_2}$.

Open Problems We see that, despite our advancements, a number of questions are still unresolved. For the study of forward-stable topologies, most of them originate from Design and Coding Theory. As we have already pointed out, one of the hardest seems to be the existence problem of Orthogonal Arrays. However, the following problems are also relevant:

- Theorem 5.3.24 demonstrates that the matrix $M^{\mathcal{T}}$ of a forward-stable topology $\mathcal{T} \in \mathbb{T}(n, C, k)$ must satisfy $M^{\mathcal{T}} \in \bigcap_{t \in [k]} \mathbb{M}(n, k, C, t)$. However, we do not know whether this intersection is always non-empty. To obtain an answer, the intersections between the classes of Packing Arrays with minimum maximum subrow frequency have to be studied.

  If non-emptiness is proven (as for MDS codes), we can study the vulnerability of topologies $\mathcal{T}$ with $M^{\mathcal{T}} \in \bigcap_{t \in [k]} \mathbb{M}(n, k, C, t)$. Here, first steps were made in Subsection 5.3.3. However, we do not yet have an exact characterization of MDS codes that are suitable to build forward-stable topologies.

- For the construction of forward-stable topologies with arbitrary peer numbers, we need Orthogonal Arrays and Packing Arrays with arbitrary row number. In this context, it would be interesting to know whether for all parameters n, k, C, t there is an $\mathrm{OA}(\hat{\mathrm{n}}(n, k, C, t), k, C, t)$ that can be decomposed into $\lfloor n/C^k \rfloor$ copies of $[C]^k$ and a single copy of a *code* M. If this was confirmed, the existence problem could be solved by studying only existence conditions of codes.

Although forward-stable topologies closely approximate optimally LoSS-stable topologies with many peers, both classes could considerably differ for small values of n. This especially applies to head topologies. Hence, an exact characterization of optimally LoSS-stable topologies is desireable. For this, new techniques of analysis have to be found. In particular, a direct adoption of the matrix repesentation used to analyze forward-stable topologies is not possible, since it heavily relies on the one-stripe-only property and the definition of forward successor sets.

Heuristics for Distributed Topology Management Last but not least, the insights we already obtained about forward- and LoSS-stable topologies should be incorporated into the existing heuristics of *distributed* topology construction mechanisms. Here, empirical studies show, that the local-cost-based system of [BSS09] already gives practical approximations for the one-stripe-only property and the successor number limitations of Lemma 5.3.2 (cmp. Section 4.5). However, the important aspect of controlling the intersections of the heads' (forward) successor sets is still unaddressed. This is an extremely challenging task since in common systems even the heads only know the *cardinality* of their successor sets and the identities of their children.

Increasing the topology knowledge of peers is mostly impractical, especially for scaling and security reasons (see, e.g., [BFGS09a]). Hence, it is necessary to make use of more indirect control factors. A key role might fall to the bootstrapping server of the streaming system which essentially assigns initial positions of newly joining nodes (afterwards they evolve due to topology dynamics). These decisions must be

based on an estimation of the current heads' (forward) successors sets. However, the storage and maintainance of this information poses the same scaling problem as before. Until better strategies are found, the purely random assignment of initial predecessors, in combination with continuous tree balancing operations, seems to be a reasonable solution. Albeit it is far from optimal, it makes very large intersections of heads' successor sets at least improbable.

Despite all the problems that arise when trying to implement optimally LoSS- or forward-stable topologies in real-world streaming systems, it nonetheless promises to result in highly improved stability properties. This is supported by the observations in [Gum11], where forward-stable topologies (in this context called 'mixed') were empirically compared with samples of other established topology classes. Through all tested attack parameter combinations, the forward-stable topologies proved to suffer considerably lower LoSS-damage than the second-best tested topologies (being optimally LiSS-stable topologies from Chapter 4 and topology snapshots of the running system [BSS09]).

6. Random-Failure-Stability

In the Chapters 4 and 5, we have considered topology formation problems with a focus on attack-stability. In particular, we have sought topologies that minimize the maximum damage of deliberate and well-planned attacks. The main difference between the goals of both chapters lay in the damage function that was applied.

For a more complete view on topology stability, we now want to deviate from the worst-case assumptions made in the previous chapters. Instead of studying targeted attacks, we will consider damages that occur due to random node failures. Since peer-to-peer-based streaming systems rely on undependable end-hosts that frequently leave the system without warning (so-called ungraceful exits), many authors consider this kind of damage as the most severe stability problem. This is reflected by the fact, that the vast majority of research on stability and resilience of peer-to-peer live streaming systems is concerned with approaches to cope with this constant loss (and re-joining) of nodes, e.g., [TJ07, TWSN08, DF10, LCC+11]. Additionally, nearly all of the resilience evaluations of the peer-to-peer live streaming systems mentioned in Section 2.1.3 focus on this kind of topology damage.

In the following, Section 6.1 will establish and motivate our model of random failures and their consequences on the streaming system. In particular, we combine the established LiSS-damage measure of counting lost source-to-peer paths with a random process choosing the failing nodes. Topologies minimizing the expected LiSS-damage of such failures are called random-failure-stable. Then, Section 6.2 identifies the random-failure-stable topologies in topology classes having only a single stripe tree. These results are extended in Section 6.3, where we will quantify the expected packet loss on *arbitrary* distribution topologies. It is determined by a weighted sum of individual node depths in all stripe trees. This observation will lead to the identification of sufficient conditions for random-failure-stable topologies. Additionally, we demonstrate that there are non-empty topology classes in which no random-failure-stable topologies exist. The section closes with an overview on topology classes for which random-failure-stable topologies are simple to construct. In contrast, we will show in the subsequent Section 6.4, that the problem of finding random-failure-stable topologies in a given topology class is strongly **NP**-complete, in general.

6.1. Random Failures and Failure-Stability

Following the discussion in Section 2.2.2, we consider topologies at a fixed snapshot in time and disregard topology dynamics. Furthermore, we focus on the study of topologies in the classes of bandwidth-restricted distribution topologies as given in Definition 2.2.5.

In contrast to the deterministic worst-case attacks studied so far, determining topology stability with regard to random node failures requires the introduction of a probabilistic model. In the considered scenario, a topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ and a number $x \in [n]$ are given, and a set $X \subseteq V$ of failing nodes with $|X| = x$ is chosen by a random process.

For peer-to-peer live streaming systems, a number of approaches were proposed to measure individual node reliability, i.e., the probability of not failing in a given time interval. Usually, they are combined with techniques aiming to bring only reliable nodes into important positions of the distribution topology. A survey of reputation systems that can be used for such approaches is given in [MGM06]. The authors of both [TJ07] and [TWSN08] propose topology management protocols considering measured node reliability. However, they restrict to single-tree topologies and evaluate their approaches using simulation studies, but without formal analysis. In [LCC+11], a topology formation problem is formulated as an Integer Linear Program. It is then used to infer parent selection rules to cope with topology dynamics. In general, the results in this line of research do not include formal analysis on failure-stable multitree distribution topologies.

In this thesis, *we will assume that each subset of failing nodes is chosen with equal probability.* Although this is an abstraction from reality, it is a good starting point giving important insights on the problem of finding topologies that are stable towards random failures. Furthermore, our general approach is transferable to other probability distributions. However, the arising stability requirements may change in dependency on the distribution used.

Given a topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ for which $X \subseteq V$ with $|X| = x$ is chosen by the random process, let us introduce a random variable $A^{\mathcal{T}}_{x,v}(X)$ for each node $v \in V$:

$$A^{\mathcal{T}}_{x,v}(X) := |\{i \in [k] \mid v \in \mathrm{succ}^{\mathcal{T}}_i(X)\}| = k - \mathrm{inc}_X(v). \tag{6.1}$$

Furthermore define

$$\begin{aligned} A^{\mathcal{T}}_x(X) &:= \sum_{v \in V} A^{\mathcal{T}}_{x,v}(X) && (6.2)\\ &= \sum_{v \in V} \sum_{i \in [k]} |\mathrm{succ}^{\mathcal{T}}_i(X) \cap \{v\}| && (6.3)\\ &= \sum_{i \in [k]} \sum_{v \in V} |\mathrm{succ}^{\mathcal{T}}_i(X) \cap \{v\}| && (6.4)\\ &= \sum_{i \in [k]} |\mathrm{succ}^{\mathcal{T}}_i(X)| = \mathrm{a}^{\mathcal{T}}(X). && (6.5) \end{aligned}$$

The random variable $A^{\mathcal{T}}_x(X)$ quantifies the LiSS-damage for a failing set X of size x chosen by the random process. Consequently, the expected value $\mathrm{E}(A^{\mathcal{T}}_x)$ will be called the *expected packet loss for x failing nodes*, or, for short, the *expected damage*. The sequence $(\mathrm{E}(A^{\mathcal{T}}_1), \ldots, \mathrm{E}(A^{\mathcal{T}}_n))$ can be used to measure the stability of topologies against random failures:

Definition 6.1.1 ***Random-Failure-Stable Topologies***
For $x \in [n]$, a distribution topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ is called *x-random-failure-stable in* $\mathbb{T}(n, \mathrm{c}, k)$, if

$$\forall \mathcal{C} \in \mathbb{T}(n, \mathrm{c}, k) \colon \mathrm{E}(A_x^{\mathcal{T}}) \leq \mathrm{E}(A_x^{\mathcal{C}}).$$

A distribution topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ is called *random-failure-stable in* $\mathbb{T}(n, \mathrm{c}, k)$, if it is x-random-failure-stable for all $x \in [n]$.

Again, we can define a corresponding topology formation problem.

Definition 6.1.2 **(x-)Failure-Stable Topology Formation Problem**
Given $n, k \in \mathbb{N}$ and an appropriate capacity function c, the *(x-)Failure-Stable Topology Formation Problem* consists in finding an (x-)random-failure-stable topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ or in determining that none exists.

Under the reasonable assumptions that c is represented as vector of $(n+1)$ entries and that k is polynomial in n, the output length (the binary representation of a topology from $\mathbb{T}(n, \mathrm{c}, k)$, e.g., an $n \times k$-matrix of predecessors) of these problems will be polynomial in the length of the input. Thus, the existence of polynomial-time algorithms is not impossible. This distinguishes formation problems of topologies in $\mathbb{T}(n, \mathrm{c}, k)$ from the formation problems of topologies in $\mathbb{T}(n, C, k)$.

We will see in Section 6.3 that every non-empty class $\mathbb{T}(n, \mathrm{c}, k)$ contains an x-random-failure-stable topology $\mathcal{T}$ for every value of $x \in [n]$ (though it may be **NP**-hard to find it). However, there exist non-empty topology classes without globally random-failure-stable topologies.

In the following, we will characterize the (x-)random-failure-stability of a topology by the number of nodes in its different depth levels. At first, we will study and identify random-failure-stable distribution topologies that consist of a single stripe tree. These results are then generalized to arbitrary topologies. Although the random-failure-stable topologies found in Sections 6.2 and 6.3 are quite intuitive, Section 6.4 will prove that the task of forming an (x-)random-failure-stable topology in arbitrary class $\mathbb{T}(n, \mathrm{c}, k)$ is indeed an **NP**-complete problem (for $x \leq n - 2$).

6.2. Failure-Stability of a Single Stripe Tree

In topology classes with $k = 1$, every topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ consists of only a single stripe tree. Under these premises, we can transform the expected packet loss as shown below. Note that $\mathrm{H}(k \mid N; M; n) = \binom{m}{k}\binom{N-M}{n-k}/\binom{N}{n}$ is the probability mass function of the hypergeometric distribution.

$$\mathrm{E}(A_x^{\mathcal{T}}) = \sum_{v \in V} \mathrm{E}(A_{x,v}^{\mathcal{T}}) \tag{6.6}$$

$$= \sum_{v \in V} \Pr(A_{x,v}^{\mathcal{T}} = 1) \tag{6.7}$$

$$= \sum_{v \in V} (1 - \mathrm{H}(0 \mid n; d(v); x)) \tag{6.8}$$

$$= n - \sum_{i=1}^{d(\mathcal{T})} \mathrm{H}(0 \mid n; i; x) \cdot |L_i(\mathcal{T})| \tag{6.9}$$

$$= n - \frac{(n-x)!}{n!} \sum_{i=1}^{n-x} \frac{(n-i)!}{(n-i-x)!} \cdot |L_i(\mathcal{T})| \tag{6.10}$$

Here, the step from Term (6.6) to (6.7) follows from the fact that $k = 1$.

Using $\mathrm{H}(0 \mid n; d(v); x)$, we can express the probability that, after the failure of x out of the n nodes with equal probability, none of v's $d(v)$ predecessors besides the source has failed. The transformation from Term (6.7) to (6.8) uses exactly the complementary event. From Term (6.9) to (6.10), we replaced $\mathrm{H}(0 \mid n; i; x)$ with its definition and resolved the binomial coefficients. Furthermore, we used the facts that $\mathrm{H}(0 \mid n; i; x) = 0$ for $i \in [n - x + 1, n]$ and that $|L_i(\mathcal{T})| = 0$ for $i > d(\mathcal{T})$.

We find that random-failure-stable topologies with $k = 1$ minimize the average node depth, since $E(A_1^{\mathcal{T}})$ has exactly this value.

$$\begin{aligned} E(A_1^{\mathcal{T}}) &= \sum_{v \in V} (1 - \mathrm{H}(0 \mid n; d(v); 1)) \\ &= \sum_{v \in V} \left(1 - \frac{n - d(v)}{n}\right) = \frac{1}{n} \sum_{v \in V} d(v) \end{aligned} \tag{6.11}$$

As another interesting observation, we also see that actual successor relationships have no influence on the value of expected damage. In particular, if two trees have an equal distribution of node numbers to node depths, they have the same expected packet loss, no matter how unbalanced one of them may be.

Now take a closer look at Equation (6.10). The values n and x cannot be influenced by the topology construction. Hence, we see that the variable part is a weighted sum over the number of nodes in the different depth levels of the tree $\mathcal{T}$. We can deduce a sufficient condition on random-failure-stable topologies consisting of just a single tree. It coincides with the intuitive approach to build the topology tree "as flat as possible".

Lemma 6.2.1
If there is a topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, 1)$ that satisfies

$$\forall \mathcal{C} \in \mathbb{T}(n, \mathrm{c}, 1)\ \forall i \in [d(\mathcal{T}) - 1] \colon |L_i(\mathcal{T})| \geq |L_i(\mathcal{C})|, \tag{6.12}$$

then the compliance with Property (6.12) is a necessary and sufficient condition for random-failure-stable topologies in $\mathbb{T}(n, \mathrm{c}, 1)$.

Proof. For each $x \in [n]$, let the sequence $w_1^x, \ldots, w_{n-x}^x$ be given by $w_j^x := \frac{(n-j)!}{(n-j-x)!}$ for $j \in [n - x]$. It holds that $\forall j \in [n - x - 1] \colon w_j^x > w_{j+1}^x > 0$.

All topologies $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, 1)$ with Property (6.12) maximize the term $\sum_{j=1}^{n-x} w_j^x |L_j(\mathcal{T})|$

for all values of $x \in [n]$. Thus, due to Equation (6.10), they minimize $E(A_x^{\mathcal{T}})$ for all $x \in [n]$ and are random-failure-stable.

Now, assume that such a $\mathcal{T}$ exists and that $\mathcal{C} \in \mathbb{T}(n, \mathrm{c}, 1)$ violates Property (6.12). It must hold that $\mathrm{d}(\mathcal{T}) \leq n-1$ and $\mathrm{d}(\mathcal{C}) \leq n$. Furthermore, there is depth $i < d(\mathcal{T})$ such that $\sum_{j=1}^{i} |L_j(\mathcal{T})| > \sum_{j=1}^{i} |L_j(\mathcal{C})|$. Since $\sum_{j=1}^{n-1} |L_j(\mathcal{T})| = n$ and $\sum_{j=1}^{n-1} |L_j(\mathcal{C})| \in [n-1, n]$, we obtain

$$\sum_{j=i+1}^{n-1} |L_j(\mathcal{C})| \geq n - 1 - \sum_{j=1}^{i} |L_j(\mathcal{C})| \geq n - \sum_{j=1}^{i} |L_j(\mathcal{T})| = \sum_{j=i+1}^{n-1} |L_j(\mathcal{T})| \tag{6.13}$$

and

$$\sum_{j=1}^{i} |L_j(\mathcal{T})| - \sum_{j=1}^{i} |L_j(\mathcal{C})| \geq \sum_{j=i+1}^{n-1} |L_j(\mathcal{C})| - \sum_{j=i+1}^{n-1} |L_j(\mathcal{T})|. \tag{6.14}$$

This leads to

$$E(A_1^{\mathcal{C}}) - E(A_1^{\mathcal{T}}) = \frac{(n-1)!}{n!} \left(\sum_{j=1}^{n-1} w_j^1 \cdot |L_j(\mathcal{T})| - \sum_{j=1}^{n-1} w_j^1 \cdot |L_j(\mathcal{C})| \right) \tag{6.15}$$

$$= \frac{(n-1)!}{n!} \left(\sum_{j=1}^{i} w_j^1 (|L_j(\mathcal{T})| - |L_j(\mathcal{C})|) + \sum_{j=i+1}^{n-1} w_j^1 (|L_j(\mathcal{T})| - |L_j(\mathcal{C})|) \right) \tag{6.16}$$

$$\geq \frac{(n-1)!}{n!} \left(w_i^1 \sum_{j=1}^{i} (|L_j(\mathcal{T})| - |L_j(\mathcal{C})|) - w_{i+1}^1 \sum_{j=i+1}^{n-1} (|L_j(\mathcal{C})| - |L_j(\mathcal{T})|) \right) \tag{6.17}$$

$$> 0. \tag{6.18}$$

Thus, $\mathcal{C}$ is not 1-random-failure-stable and not random-failure-stable. □

Failure-stable Trees Clearly, without any node degree restrictions, a source-rooted star will always be the only random-failure-stable topology consisting of one tree. However, also in the presence of capacity restrictions, the formation of a topology $\mathcal{T}$ complying with the Property (6.12) is always possible if $\mathbb{T}(n, \mathrm{c}, 1)$ is non-empty. This can be done using the greedy approach of iteratively assigning the nodes V in order of non-increasing capacity to a free tree position with smallest depth. This way, the depth levels are filled consecutively. In particular, in each but the last level the capacity of nodes in the former level is fully utilized while the capacity for nodes in the next depth level is maximized. It is possible to deviate from the node order in the last two depth levels, as long as the capacity of all nodes in the other levels is exhausted [Rö10].

Note that many peer-to-peer streaming systems that build only a single distribution tree intuitively rely on the tree topologies sketched above. For example, one of the three basic design principles of FatNEMO [BLB+04] is that "higher degree nodes should be placed higher up in the tree". Other approaches which explicitely aim at building

"short and wide" trees are [PWCS02] and [GA04].

Additionally, the authors of [TJ07] introduce a random model for failure stability which is similar to ours and advise to build bandwidth-ordered trees after conducting a simulation study on the model.

Since we have seen that every non-empty class $\mathbb{T}(n, \mathrm{c}, 1)$ contains a tree with Property (6.12), we obtain the following corollary.

Corollary 6.2.2
A distribution topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, 1)$ is random-failure-stable if and only if it has Property (6.12).

In the following section, we will generalize these results to multitree topologies. However, we will see that the existence of random-failure-stable topologies is no longer guaranteed for $k > 1$.

6.3. Failure Stability of Distribution Topologies

Due to the linearity of expectation, we can lead back the expected packet loss on arbitrary distribution topologies to that of their stripe trees:

$$\begin{aligned}
\mathrm{E}(A_x^{\mathcal{T}}) &= \sum_{v \in V} \mathrm{E}(A_{x,v}^{\mathcal{T}}) && (6.19)\\
&= \sum_{v \in V} \sum_{T_j \in \mathcal{T}} \mathrm{E}(A_{x,v}^{T_j}) && (6.20)\\
&= \sum_{T_j \in \mathcal{T}} \mathrm{E}(A_x^{T_j}) && (6.21)\\
&= kn - \sum_{i=1}^{d(\mathcal{T})} \mathrm{H}(0 \mid n; i; x) \sum_{T_j \in \mathcal{T}} |L_i(T_j)| && (6.22)\\
&= kn - \sum_{i=1}^{d(\mathcal{T})} \mathrm{H}(0 \mid n; i; x) |L_i(\mathcal{T})| && (6.23)\\
&= kn - \frac{(n-x)!}{n!} \sum_{i=1}^{n-x} \frac{(n-i)!}{(n-i-x)!} \cdot |L_i(\mathcal{T})| && (6.24)
\end{aligned}$$

Consequently, the value $\mathrm{E}(A_x^{\mathcal{T}})$ corresponds to kn minus a weighted sum of individual node depths in the trees of $\mathcal{T}$. For each fixed value of $x \in [n]$, the weight factors for each depth are fixed, too. Hence, there will always be a topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ minimizing $\mathrm{E}(A_x^{\mathcal{T}})$ by having an optimal depth distribution for these weights.

We can generalize Lemma 6.2.1 to arbitrary k.

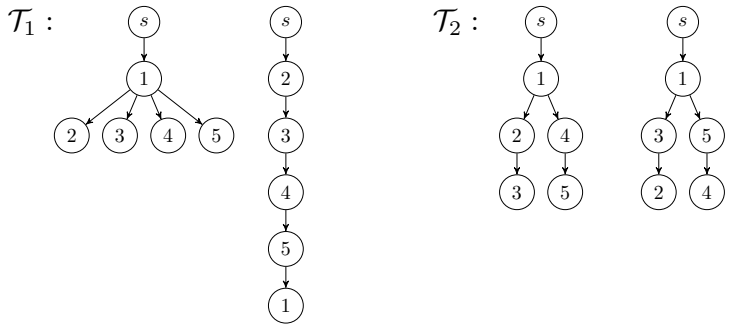


Figure 6.1.: Example topologies $\mathcal{T}_1$,$\mathcal{T}_2 \in \mathbb{T}(5, \mathrm{c}, 2)$ with $\mathrm{c}(s) = 2$, $\mathrm{c}(1) = 4$ and $\mathrm{c}(v) = 1$ for $v \in [2, 5]$.

Lemma 6.3.1
If there is a distribution topology $\mathcal{T} \in \mathbb{T}(n, \mathrm{c}, k)$ that satisfies

$$\forall \mathcal{C} \in \mathbb{T}(n, \mathrm{c}, k)\ \forall i \in [d(\mathcal{T}) - 1] \colon |L_i(\mathcal{T})| \geq |L_i(\mathcal{C})|, \tag{6.25}$$

then the compliance with this property is a sufficient and necessary condition for random-failure-stable topologies in $\mathbb{T}(n, \mathrm{c}, k)$.

The proof is analogue to that of Lemma 6.2.1, but is referencing Equation (6.24).

Note that topologies with Property (6.25) would also be optimal for other probability distributions of node failure, as long as the probability that a node's predecessor (including the node itself) has failed is a monotonously increasing function depending *only* on the node's depth.

In contrast to the case $k = 1$, for $k > 1$ the existence of topologies with Property (6.25) is no longer guaranteed. An example is given by the class $\mathbb{T}(5, \mathrm{c}, 2)$ with $\mathrm{c}(s) = 2$, $\mathrm{c}(1) = 4$ and $\mathrm{c}(v) = 1$ for $v \in [2, 5]$. Since the source can support exactly one head per stripe, all topologies in $\mathbb{T}(5, \mathrm{c}, 2)$ have the same number of heads. To maximize the number of nodes in depth level two, we have to use two different heads, one of which must be the node 1. Since the remaining nodes have equal capacities, any can be adopted as second head. Topology $\mathcal{T}_1$ in Figure 6.1 has these properties. However, such a one-stripe-only assigment of heads prevents the second stripe to fan out near the source. Consequently, it forces $\mathcal{T}_1$ to become very deep, with few peers in its deeper levels. In contrast, topology $\mathcal{T}_2$ in Figure 6.1 only has depth 3 and it holds that $|L_3(\mathcal{T}_1)| \leq |L_3(\mathcal{T}_2)|$. Hence, in $\mathbb{T}(5, \mathrm{c}, 2)$ the maximization of the depth levels 2 and 3 are conflicting goals. Since $\mathbb{T}(5, \mathrm{c}, 2)$ does not contain topologies of depth 2, no topology in this class has Property (6.25).

The lack of such topologies can lead to the inexistence of random-failure-stable topologies. Let $\mathbb{T}(n, \mathrm{c}, k)$ be a non-empty class *without a topology having Property* (6.25) and let $\hat{d} = \min_{\mathcal{T} \in \mathbb{T}(n,\mathrm{c},k)} d(\mathcal{T})$. Then there have to be $a, b \in [n]$ with $a < b \leq \hat{d}$ and

topologies $\mathcal{T}_a, \mathcal{T}_b \in \mathbb{T}(n, \mathrm{c}, k)$ maximizing the number of nodes in either depth a or b. Depending on the depth distributions of $\mathcal{T}_a$ and $\mathcal{T}_b$, and the proportions of the elements of sequence $(\mathrm{H}(0|n; i; x))_{i\in[n-x]}$ for different x, it is then possible that $\mathcal{T}_b$ is $(n-b)$-random-failure-stable while $\mathcal{T}_a$ is not, even though $\mathcal{T}_b$ has less nodes in depth level a.

Again, Figure 6.1 gives an example. For $x = 3$, only the depth levels one and two determine the expected packet loss of a topology in $\mathbb{T}(5, \mathrm{c}, 2)$ (cmp. Equation (6.24)). The depicted topology $\mathcal{T}_1$ and all other topologies with its depth distribution are 3-random-failure-stable in $\mathbb{T}(5, \mathrm{c}, 2)$ (e.g., $E(A_3^{\mathcal{T}_1}) = 8.7$ vs. $E(A_3^{\mathcal{T}_2}) = 8.8$). However, its depth makes $\mathcal{T}_1$ suboptimal for $x = 1$. Here, topology $\mathcal{T}_2$ is a 1-random-failure-stable topology (e.g., $E(A_1^{\mathcal{T}_1}) = 4.8$ vs. $E(A_1^{\mathcal{T}_2}) = 4.4$).

Random-Failure-Stable Topologies for Special Topology Classes There are a number of classes $\mathbb{T}(n, \mathrm{c}, k)$ where random-failure-stable topologies are easy to find. If $\mathrm{c}(s) \geq kn$, a topology of star-like stripe trees rooted at s is optimal due to Lemma 6.3.1. Furthermore, if $\mathrm{c}(s) < kn$ and only the source has limited capacity, each topology with fully exhausted source capacity and maximum depth 2 will be optimal.

Classes with higher practical relevance have a common capacity value D for all peers V. Under the assumption that $\mathbb{T}(n, \mathrm{c}, k) \neq \emptyset$, it is then possible to use complete D-ary subtrees rooted at the $\mathrm{c}(s)$ heads. This maximizes the number of nodes in each but the last level of all subtrees. Distributing the number of heads among the stripes as equal as possible (i.e., $\forall i \in [k]\colon \lfloor \mathrm{c}(s)/k \rfloor \leq |L_1(T_i)| \leq \lceil \mathrm{c}(s)/k \rceil$), we then obtain a topology with depth $\lceil \log_D(\lceil n/\lfloor \mathrm{c}(s)/k \rfloor \rceil \cdot (D-1) + 1) \rceil$ that satisfies Property (6.25).

Related Work: Enabling Forward Error Correction The authors of [DF10] give a highly detailed and complex probabilistic model of the packet loss in distribution topologies when Forward Error Correction encoding is applied. Then, they introduce extensive, simplifying assumptions and evaluate their model by simulation. In contrast to our results without Forward Error Correction, their simulations indicate that FEC mechanisms additionally profit from nodes whose depth has only limited (e.g. constant) deviations between the stripe trees. This is somewhat suprising since such a topology layout disagrees with Property (6.25) in many topology classes. An analytical investigation of this phenomenon and the determination of optimal trade-offs promises to be an interesting direction for future research.

6.4. Complexity of Finding Random-Failure-Stable Topologies

Although the failure-stable topologies identified in the Sections 6.2 and 6.3 had quite an intuitive layout, we now show that the Failure-Stable Topology Formation Problem and most of its x-restricted versions are **NP**-complete in general.

Theorem 6.4.1
The Failure-Stable Topology Formation Problem is **NP**-hard.

Proof. To prove **NP**-hardness, we will use a polynomial-time reduction from the strongly **NP**-complete problem 3-Partition ([GJ79] problem SP15).

Definition 6.4.2 ***3-Partition***
Given a weight vector $(a_1, \ldots, a_w)$ with $w = 3m$, $m, A \in \mathbb{N}$, $a_i \in \mathbb{N}$, and $\sum_{1 \leq i \leq w} a_i = A \cdot m$, decide whether a partition of $[w]$ into sets $J_1, \ldots, J_m$ exists, such that $\forall i \in [m]\colon \sum_{j \in J_m} a_j = A$.

The 3-Partition problem remains **NP**-complete when we restrict the values a_i such that $A/4 < a_i < A/2$, thus enforcing $\forall i \in [m]\colon |J_i| = 3$ [GJ79]. Since it is *strongly* **NP**-complete, we can also restrict A to be upper bounded by a polynomial in w without losing **NP**-completeness. In the following we assume that both these restrictions apply.

Given a weight vector $(a_1, \ldots, a_w)$, the reduction function f produces an instance of the Failure-Stable Topology Formation Problem with topology class $\mathbb{T}(3 + 2A, \mathrm{c}, m)$ (i.e., $V = [3 + 2A]$) and capacity function

$$\mathrm{c}(v) = \begin{cases} w & \text{, if } v = s \\ 2a_v & \text{, if } v \leq w \\ 1 & \text{, else.} \end{cases} \tag{6.26}$$

Under the above restrictions, computing this mapping and returning c in a pair-wise representation needs time at most polynomial in the length of the input $(a_1, \ldots, a_w)$.

Define $V_w := [w]$ and let $D := [3+2A] \setminus V_w$ be the set of *dummy nodes*. The instances constructed by f have the following property:

Claim 6.4.3
There is a 3-Partition $J_1, \ldots, J_m$ for weight vector $(a_1, \ldots, a_w)$ if and only if a solution $\mathcal{T}$ for the Failure-Stable Topology Formation Problem on $f((a_1, \ldots, a_w))$ exists and satisfies

$$\forall T_i \in \mathcal{T}\colon |L_2(T_i)| = 2A.$$

Proof. "Only-If": Assume that the 3-Partition solution $J_1, \ldots, J_m$ exists. Then a topology $\mathcal{T} \in f((a_1, \ldots, a_w))$ can be built with $H_i^{\mathcal{T}} = J_i$ for $i \in [m]$ in which, per stripe $T_i \in \mathcal{T}$, all $2A$ remaining nodes $V \setminus H_i^{\mathcal{T}}$ form the second depth level. Since $\forall i \in [m]\colon \sum_{v \in J_i} \mathrm{c}(v) = \sum_{v \in J_i} 2a_v = 2A$, the capacity of the heads allows such a construction in each stripe. See Figure 6.2 for an example. Since the source capacity $\mathrm{c}(s) = w$ is utilized completely, the maximum possible number of nodes reside in depth level 1. Furthermore, it holds that $d(\mathcal{T}) = 2$. Due to Lemma 6.3.1, $\mathcal{T}$ must be random-failure-stable in $f((a_1, \ldots, a_w))$.

"If": Assume that a random-failure-stable topology $\mathcal{T}$ for $f((a_1, \ldots, a_w))$ has $2A$ nodes of depth 2 per stripe. This requires nodes in depth level 1 with a capacity sum of at least

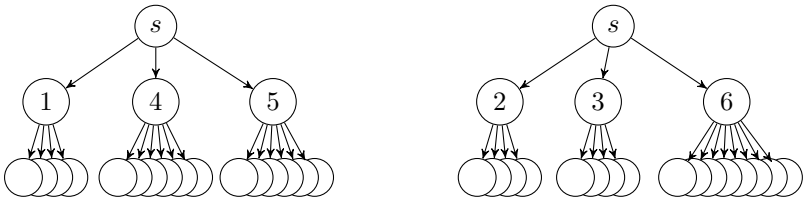


Figure 6.2.: A random-failure-stable topology in $f((a_1, \ldots, a_6))$ for 3-Partition input $(a_1, \ldots, a_6) = (2, 2, 2, 3, 3, 4)$ with $A = 8$ and a 3-Partition $J_1 = \{2, 3, 3\}$, $J_2 = \{2, 2, 4\}$.

$k \cdot 2A = m \cdot 2A = \sum_{u \in V_w} \mathrm{c}(u)$. Since $\mathrm{c}(s) = w = |V_w|$ and $\forall u \in V_w, v \in D\colon \mathrm{c}(u) > \mathrm{c}(v)$, we must have $H^{\mathcal{T}} = V_w$ and the stripe head sets $H_i^{\mathcal{T}}$ must be a partition of V_w. Additionally, the heads of each stripe must have *exactly* capacity $2A$. Otherwise, due to the total capacity of $m \cdot 2A$, there would be a stripe with capacity *less* than $2A$. This would contradict our assumptions about $\mathcal{T}$. Hence, by definition of c, the sets $J_i := H_i^{\mathcal{T}}$ for $i \in [m]$ are a 3-Partition solution for $(a_1, \ldots, a_w)$.

Finally, let us show, that if *any* random-failure-stable topology $\mathcal{T}$ for $f((a_1, \ldots, a_w))$ has $\forall T_i \in \mathcal{T}\colon |L_2(T_i)| = 2A$, then *every* random-failure-stable topology in $f((a_1, \ldots, a_w))$ has this feature. For this, assume that $\mathcal{C} \in f((a_1, \ldots, a_w))$ is random-failure-stable and has a stripe j without $2A$ nodes in depth 2. Due to Lemma 6.3.1 and the existence of $\mathcal{T}$, topology $\mathcal{C}$ must have w nodes in depth level 1, must have depth 2, and furthermore have $|L_2(\mathcal{C})| = m2A$. W.l.o.g. we can assume that stripe j has $|L_2(T_j)| > 2A$. Then, it holds that $|H_j^{\mathcal{T}}| \leq 2$ with $\sum_{v \in H_j^{\mathcal{T}}} \mathrm{c}(v) > 2A$. However, this is a contradiction with our assumption that $\forall i \in [w]\colon A/4 < a_i < A/2$. □

□

Now study the complexity of the x-Failure-Stable Topology Formation Problem. For $x = n$, all n nodes fail. Consequently, all topologies in a class $\mathbb{T}(n, \mathrm{c}, k)$ are n-random-failure-stable. For $x = n - 1$, due to Equation (6.24), every topology maximizing $|L_1(\mathcal{T})|$, i.e., completely utilizing the source capacity, is $(n-1)$-random-failure-stable. Hence, if k is polynomial in n, the x-Failure-Stable Topology Formation Problem for $x \geq n - 1$ is in **P**.

We have seen in Claim 6.4.3 that there is a 3-Partition for $(a_1, \ldots, a_w)$ if and only if there is a random-failure-stable topology $\mathcal{T}$ in $f((a_1, \ldots, a_w))$ that has $2A$ nodes in depth level 2 of *each* stripe. By definition, such a topology $\mathcal{T}$ is also x-random-failure-stable for all $x \in [n]$.

Furthermore, the proof of Claim 6.4.3 has shown that the head sets of every topology in $f((a_1, \ldots, a_w))$ that has the maximum possible number of w nodes in depth 1 and the maximum possible number of $m \cdot 2A$ nodes in depth level 2 induce a 3-Partition of $(a_1, \ldots, a_w)$. Since, for $x \in [n-2]$, these are also the x-random-failure stable topologies in $f((a_1, \ldots, a_w))$ (again cmp. Equation (6.24)), the complexity result

of Theorem 6.4.1 also applies to the x-Random-Failure-Stable Topology Formation Problem with $x \leq n-2$.

Corollary 6.4.4
The x-Failure-Stable Topology Formation Problem is **NP**-hard for $x \in [n-2]$.

Furthermore, each x-Failure-Stable Topology Formation Problem has a corresponding **NPO** problem (see Definition 3.2.4): The input set is given by the set of all possible classes $\mathbb{T}(n, \mathrm{c}, k)$, possible solutions are topologies from these classes, and membership in a class can be checked by an $O(kn)$-time tree traversal. The value of a solution $\mathcal{T}$ is $E(A_x^{\mathcal{T}}) \cdot \frac{n!}{(n-x)!}$ and can be checked in polynomial time. Note that the factor $\frac{n!}{(n-x)!}$ is used to scale $E(A_x^{\mathcal{T}})$ to a natural number. This is necessary to satisfy the requirements on a value funtion. Finally, the optimization goal is a minimization. The search version looking for an optimal solution to this **NPO** problem is the x-Failure-Stable Topology Formation Problem.

If restricted to topology classes $\mathbb{T}(n, \mathrm{c}, k)$ in which random-failure-stable topologies exist, the Failure-Stable Topology Formation Problem has a similar corresponding **NPO** problem: The input set consists of all topology classes in which random-failure-stable topologies exist, solutions are topologies from the input class and their membership in the class can be checked in polynomial time. The value of a solution $\mathcal{T}$ is given by $\sum_{x=1}^{n} E(A_x^{\mathcal{T}}) \cdot \frac{n!}{(n-x)!}$. Again, the optimization goal is a minimization. Since random-failure-stable topologies minimize $E(A_x^{\mathcal{T}})$ for all values of $x \in [n]$, they are exactly the optimal solutions to this optimization problem.

Corollary 6.4.5
Each x-Failure-Stable Topology Formation Problem with $x \in [n-2]$ is **NP**-complete. The Failure-Stable Topology Formation Problem on topology classes containing random-failure-stable topologies is **NP**-complete.

6.5. Summary

In this chapter, we deviated from the approach of the former chapters by not taking a worst-case viewpoint on topology stability. Instead, we considered the *expected packet loss* on a topology, when the failing node sets are chosen by a random process.

Assuming a uniformly distributed failure probability, we defined x-random-failure-stable topologies as the topologies minimizing the expected packet loss for x failing nodes. If a topology is x-random-failure-stable for all values of $x \in [n]$, it was called random-failure-stable.

In Section 6.2, we then identified all random-failure-stable topologies in topology classes allowing only a single stripe. Building on these results, in Section 6.3, we characterized the expected packet loss on multitree topoplogies. Its value is determined by a weighted sum of node depths over all stripes of a topology. We identified sufficient requirements for random-failure-stable topologies, but also highlighted that there are non-empty topology classes without random-failure-stable topologies. In these classes,

the sets of x-random-failure-stable topologies for different values of $x \in [n]$ do not intersect.

After a short review of topology classes in which random-failure-stable topologies are easy to find, we contrasted in Section 6.4 by proving that it is actually an **NP**-hard problem to find random-failure-stable topologies for arbitrary given bandwidth-restricted topology classes.

Since it holds that

$$E(A_1^{\mathcal{T}}) = \sum_{T_i \in \mathcal{T}} E(A_1^{T_i}) = \frac{1}{n} \sum_{v \in V} \sum_{T_i \in \mathcal{T}} d_{T_i}(v), \tag{6.27}$$

1-random-failure-stable and random-failure-stable topologies are always the topologies in $\mathbb{T}(n, \mathrm{c}, k)$ having a minimum average node depth. Since node depth is typically proportional to the delay of the streaming data when it arrives at a node, our complexity results are equally relevant in situations where not random-failure-stability but efficiency and quality of the streaming service are the central optimization goals.

Finally, note that it is thinkable to interprete the (x-)Failure-Stable Topology Formation Problem as one of finding k degree-bounded spanning trees in a complete graph that minimize certain node distances. One could then try to obtain additional insights into the problem by utilizing the quite significant research results on problems of constructing degree-bounded spanning trees on (weighted) graphs. Here, studied optimization goals include minimizing tree depth/latency/diameter [CGM83, KLS03, HA07], minimizing maximum node degree ([GJ79] problem ND1) and tree costs [SL07] for one spanning tree, respectively. There are no results known to the author that consider problems where $k > 1$ degree-bounded spanning trees have to be found.

However, the allowance of arbitrary graphs and the introduction of edge weights *heavily* change the character of the studied problems. These additions can considerably restrict the set of possible solutions and typically the choice of the graph or weighting determines the hardness of the problems. Hence, the studies on these problems follow very different premises. Although, this line of research may inspire us to extend the Failure-Stable Topology Formation Problem to consider an underlying graph, the existing research results scarcely add to our knowledge about the problem in its current formulation.

7. Conclusion and Outlook

This is the last chapter of this thesis. Section 7.1 summarizes and correlates the obtained results. Furthermore, it is pointed out that the identified topology classes can also be applied in other areas than peer-to-peer live streaming systems. Section 7.2 lists open problems and indicates possible directions of future research.

7.1. Conclusion

In this thesis, we studied how the stability of peer-to-peer live streaming systems is influenced by their distribution topologies. In particular, we investigated properties of distribution topologies that help minimizing the consequences of destructive events on such systems. Furthermore, we examined to what degree these properties can be checked or established in polynomial time.

The considered types of destructive events were attacks on and failures of peer nodes. The consequences of both can be modeled by a removal of peers from the distribution topology. However, they differ in the way the peer set is chosen. While the peer selection process for failures can be seen as a random process, attacks are planned by a malicious entity trying to maximize damage. We set a special focus on attack-stability, but obtained results on failure-stability as well.

Most of our results were obtained by identifying and analyzing optimization problems that reflect the considered stability aspects. The mathematical model underlying this approach is a generalization of the model from [BSS09].

Damage Measures To quantify the consequences of attacks, we considered three different damage measures. The LiSS-damage measure accounts for the system-wide number of disturbed source-to-peer paths. In contrast, the LoSS- and FEC-LoSS-damage measures count the number of nodes that receive less than a given fraction of stripes. In that, they model the user-perceived quality of the streaming service. While the LoSS-damage measure assumes a stream encoding based on Multiple Description Coding, the FEC-LoSS-damage measure assumes Forward Error Correction codes.

Approximability of Attacker Problems For each of these damage measures, we investigated the computational complexity and approximability of planning resource-efficient attacks. Given a distribution topology, certain attack parameters, and a damage threshold, such a planning task consists in finding a corresponding attack of minimum cardinality that creates at least the required damage. We obtained inapproximability results relying on the assumption that $\mathbf{P} \neq \mathbf{NP}$. On input topologies having k stripes and n peers, respectively, we proved inapproximability bounds of $\Theta(\log k)$ and $\Theta(\log n)$

for both LiSS- and LoSS-damage measure. Furthermore, we identified a logarithmic approximation algorithm for the LiSS-problem, while for the LoSS-problem this was only possible for restrictions to certain inputs. Considering FEC-LoSS-damage, we could show considerably higher inapproximability bounds with factors of $2^{\log^{1-o(1)}\Theta(k)}$ and $2^{\log^{1-o(1)}\Theta(\sqrt{n})}$, respectively. These results demonstrate the influence of system parameters like stripe number and stream encoding on the hardness of planning resource-efficient attacks.

Optimally LiSS-stable Topologies We studied distribution topologies minimizing the maximum LiSS-damage achievable by attacks of each possible cardinality. For this, we first reviewed existing results from [BSS09]. Here, the damage sequence $(\delta_i^{C,k})_{1\leq i\leq Ck}$ was introduced together with a greedy polynomial-time attack algorithm. On arbitrary topologies, it returns attacks X with a LiSS-damage lower bounded by $\sum_{i=1}^{\min(|X|,Ck)}\delta_i^{C,k}$. Furthermore, the Cluster Topologies were identified. On these restrictive but simple-to-constuct topologies, the LiSS-damage of attacks is upper-bounded by exactly the same sum over the damage sequence. Consequently, both a damage-based characterization of optimally LiSS-stable topologies and a first subclass of such topologies were found.

Subsequently, we studied necessary conditions on optimally LiSS-stable topologies. Slightly strengthening these requirements, we stated a set of rules that define a subclass, which is larger and less restrictive than the Cluster Topologies.

One of these rules required that the supply relationships between heads correspond to optimally LiSS-stable head topologies. Therefore, we turned to study these special topologies. For this, we developed an adapted characterization of LiSS-stability based on dependency graphs. Additional to the necessary requirements already known, we obtained the Stability Requirements. The identification of the Line Graph Criterion provided us with a large and generic class of optimally LiSS-stable head topologies. Further results allowed to efficiently recognize optimally LiSS-stable head topologies with few stripes or small connected components in their dependency graphs.

Then, we investigated the computational complexity of deciding whether a given distribution topology is optimally LiSS-stable. We presented a result of [Bri08] confirming that this problem is **coNP**-complete. It emphasizes the practical importance of Cluster Topologies and rule-based topologies as large subclasses for which membership tests are in **P**. Especially the latter additionally feature low requirements on peer bandwidths and provide high flexibility of possible topology layouts. A multitude of head topologies is available and if $n \gg Ck$, local topology decisions of a majority of peers are not limited by the rule set. Due to this applicability, we sketched possible heuristics to approximate both Cluster Topologies and rule-based topologies using distributed topology management mechanisms.

Optimally LoSS-stable Topologies and Forward-Stable Topologies We studied distribution topologies minimizing the maximum LoSS-damage achievable by attacks of each possible parameter combination. Since LoSS- and FEC-LoSS-damage measure have equal values on topologies of depth at most 2, the obtained results are also relevant

when FEC stream encoding is applied. In a first step, we deduced basic requirements for optimally LoSS-stable topologies. Then, we observed that the LoSS-damage measure can be seen as a superimposition of two independent types of damage. For practically relevant cases where $n \gg Ck$, one of them dominates LoSS-damage. It was called forward-damage and we focused on finding topologies minimizing this kind of damage. Again, we identified basic requirements on such forward-stable topologies and ensured that they are reconcilable with the observed requirements on optimally LoSS-stable topologies.

If the found requirements are satisfied, the forward-stability of a topology is determined by the forward successor sets of its heads. In particular, there is a convenient matrix representation of these sets. We found out that in forward-stable topologies these matrices must be Orthogonal Arrays and specific Packing Arrays. If such a matrix has strength t, the corresponding topology was shown to be forward-stable against attacks where each attacked node forwards in at least one of t stripes. The observed matrix characterization allows to draw connections with existing theory on FEC codes. We reviewed relevant results and pointed out their importance for the study of forward-stable topologies.

Finally, we investigated existence conditions for the required matrix types. This led to the insight that if an efficient construction for forward-stable distribution topologies was found, this would also solve long-standing open problems in Design and Coding Theory. In particular, it would lead to the efficient identification of extremal parameters of Orthogonal Arrays and would prove or disprove the MDS Conjecture.

Random-Failure-Stable Topologies We studied distribution topologies minimizing the expected LiSS-damage, when failing sets of peers are chosen uniformly at random. These topologies were called random-failure-stable. By analyzing the formula of expected LiSS-damage, we obtained a sufficient requirement for these topologies. It confirms the frequently applied intuitive approach of building topologies that minimize average node depth. For classes of multitree topologies, we pointed out that the existence of random-failure-stable topologies is not guaranteed. Furthermore, we showed that if they exist, the problem of finding random-failure-stable topologies is **NP**-complete.

Conditionality of Different Stability Goals Recapitulating the requirements for each considered stability goal, we can investigate whether there are goals that imply or contradict each other. However, this is complicated by the fact that our knowledge about optimal topologies differs for each different stability goal. Furthermore, we considered different bandwidth restrictions for the study of attack- and failure-stable topologies, respectively.

Generally, we could observe that all studied types of stable topologies need to utilize the available source bandwidth completely.

However, the stable topologies for each stability goal have unqiue combinations of additional properties. All attack-stable (i.e., optimally LiSS-stable, optimally LoSS-stable, or forward-stable) topologies profit from the use of a maximum number of disjoint heads. Furthermore, they set upper bounds on the successor numbers of their peers.

$\subseteq$	opt. LiSS	opt. LoSS	forward	random-failure
opt. LiSS	✓	$\times^1$	$\times^1$	$\times^2$
opt. LoSS	?	✓	?	$\times^2$
forward	$\times^3$	?	✓	$\times^2$
random-failure	$\times^4$	$\times^4$	$\times^4$	✓

Remarks:
[1] No requirements on successor set intersections.
[2] Average node depth minimization not required.
[3] No requirements on successor relationships between heads.
[4] No limits on successor numbers.

Figure 7.1.: Subclass relationships between classes of stable topologies.

Again, these particularly apply to heads. Optimally LiSS-stable topologies additionally restrict the supply relationships between their heads to certain patterns. Optimally LoSS-stable topologies prohibit unidirectional successor relationships between the same pair of nodes in more than one stripe. Forward-stable topologies even strengthen this requirement by demanding, for each peer, empty forward successor sets in all but one stripe. Both of the latter types of stable topologies additionally depend on specific intersection structures of the forward successor sets of their peers. Finally, random-failure-stable topologies depend on a strict minimization of average node depth.

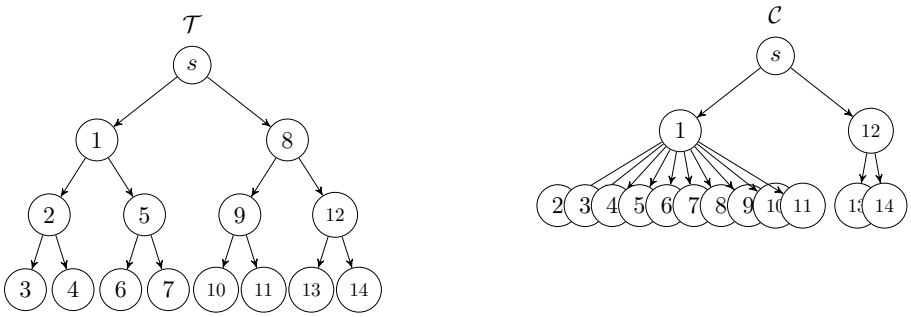
Consequently, the existence of actual subclass relationships between most of the studied classes of stable topologies can be denied (see Figure 7.1).

An exception are several unknown relationships of the optimally LoSS-stable topologies. This is due to the fact, that the exact requirements for such topologies are yet unknown. We demonstrated that the general layout of optimally LoSS-stable topologies with $n \gg Ck$ nodes must be very similar to that of forward-stable topologies. Furthermore, we have shown that upper-bounding achievable LiSS-damage also results in upper bounds on LoSS-damage. However, it is still possible that there are optimally LoSS-stable topologies (especially small ones) that are neither optimally LiSS-stable nor optimally forward-stable.

Compatibility of Different Stability Goals There are only few cases in which we can show that the intersections of different classes of stable topologies are empty.

One of these cases was given in Section 5.4. There, we demonstrated the existence of parameters for a topology class $\mathbb{T}(n, C, k)$ enforcing that there is neither a Cluster Topology nor a rule-based topology in $\mathbb{T}(n, C, k)$ that is forward-stable. Furthermore, Cluster Topologies often conflict with optimal LoSS-stability. For rule-based topologies, this is still unknown.

Another case can be observed for intersections between attack-stable and random-failure-stable topology classes. Such results must be interpreted with care, since the latter are defined on topology classes $\mathbb{T}(n, \mathrm{c}, k)$ that allow arbitrary bandwidth restrictions for peers. In particular, it is possible to find bandwidth-restricted topology



(a) Attack-stable topology.

(b) Random-failure-stable topology.

Figure 7.2.: An attack-stable topology $\mathcal{T}$ vs. a random-failure-stable topology $\mathcal{C}$ in $\mathbb{T}(14, \mathrm{c}, 1)$ with $\mathrm{c}(s) = 2$, $\mathrm{c}(1) = 10$ and $\mathrm{c}(v) = 2$ for $v \in [2, 14]$.

classes $\mathbb{T}(n, \mathrm{c}, k)$ in which attack-stable topologies have properties different to the ones identified for our standard class $\mathbb{T}(n, C, k)$. In other bandwidth-restricted classes, our familiar attack-stable topologies exist, but there are no random-failure-stable topologies (see the example in Figure 6.1, where topology $\mathcal{T}_1$ is attack-stable).

However, the optimization of attack-stability and failure-stability can also conflict if optima for both stability goals exist. A corresponding example is given in Figure 7.2. In the considered topology class, the limitation of successor numbers to $\delta_1^{C,k} = 7$ impedes a minimization of average node depth and vice versa.

Relevance and Further Applicability of Results The identified requirements on stable distribution topologies give mathematically sound guidelines for the topology management of peer-to-peer live streaming systems. The demonstrated limitations, both for attackers and the efficient construction of LoSS-, forward-, and random-failure-stable topologies, can support the evaluation of trade-offs between threat and costs of safeguarding. Consequently, they help in choosing appropriate stability goals that match a streaming system's intended use.

The studied optimization problems provide practically motivated representatives from different complexity and approximability classes. Furthermore, we found previously unknown applications of Design and Coding Theory.

The applicability of the identified stable distribution topologies is not restricted to peer-to-peer live streaming systems. Instead, such stable topologies are needed in other situations as well. Examples can be found in technically different implementations of information multicast, but also in the design of critical physical infrastructures (e.g., for the distribution of oil, gas, and electric power).

Furthermore, our damage measures remain relevant when information shall not be distributed but aggregated. This leads to possible applications in sensor networks [ASSC02]. In these systems, a large number of sensor nodes individually generate data which is decentrally aggregated and forwarded to a central sink. Depending on the

stability requirements, environmental conditions and type of sensor data, LiSS-, LoSS-, or random-failure-stability can be desirable properties of the emerging communication topologies.

7.2. Outlook

Our research can be continued in several directions.

Improved Analysis of the Studied Problems The results on a number of problems we have studied in this thesis can be extended even further.

When considering attacker problems, the identified approximability bounds for the LoSS- and the FEC-LoSS-problem are not yet tight. Here, the identification of improved bounds would give a more exact impression of the influence of stream encoding on the hardness of planning resource-efficient attacks.

In our study of optimally LiSS-stable topologies, we could not completely resolve the computational complexity of the LiSS-Stability Decision Problem for head topologies. However, the head topologies for which this problem is not yet known to be in **P** have very specific properties. A further result in this direction promises to increase the number of head topologies applicable in rule-based optimally LiSS-stable topologies.

Additionally, more detailed requirements on optimally LoSS-stable topologies should be investigated. While the obtained results already provide a good approximation when topologies possess a large number of peers, our knowledge about optimally LoSS-stable head topologies still needs to be improved.

Distributed Construction of Attack-Stable Topologies Another task that is constantly present in the study of stable distribution topologies is the research for suitable distributed mechanisms to implement our theoretical results in practical systems. We have seen that there are already peer-to-peer live streaming systems aiming to form optimally LiSS-stable topologies [BSS09, Fis12]. Their approaches are heuristical but practical. A logical next step would be the integration of additional mechanisms to improve the LoSS- or forward-stability of these systems. First ideas were given in Section 5.4. However, this is an ambitious project since constructing optimal topologies involves finding Orthogonal Arrays of maximum strength. As we have seen, even an efficient, centralized approach solving this subproblem is currently unknown.

Model Extensions Finally, it is possible to analyze extensions of the models used in this thesis. Such new models could reflect selected aspects more accurately or integrate additional features of peer-to-peer live streaming systems.

First steps in both directions have already been made. The model of [Hol10] explicitely focused on the consequences of bandwidth-exhaustion attacks. In particular, attacked nodes were no longer removed from the topology but the loss probability of packets traversing them was increased depending on the load situation on their access link. While this approach revealed interesting dynamic effects, the highly increased complexity led to difficulties in its analysis.

Another approach was followed by the author in [FGKS11]. Our model was generalized to fit for IPTV systems distributing a large number of stripes from which peers can choose to receive arbitrary subsets. Thus, each stripe can have a different set of receiving nodes. This makes the construction of attack-stable distribution topologies even more complex. The publication provided efficient methods to find topologies reducing maximum LiSS-damage to values near the possible minimum.

Further extensions of our model could consider peer-to-peer live streaming systems following a hybrid instead of push-based approach (cmp. Subsection 2.1.3). In particular, it would be possible to generalize our stripe trees into directed acyclic graphs. Then, a peer v only loses a stripe if all $s \to v$-paths in the corresponding graph fail. A problem of such systems is that bandwidth must be reserved for the additional links. This model would allow to study possible stability gains of such hybrid topologies and investigate bandwidth-efficient implementations.

It is also possible to study concepts of transitive damage. Extending the LoSS- and FEC-LoSS-damage measures, such approaches take into account that damaged nodes are highly motivated to leave the streaming system. This may lead to further disturbances. The consideration of such transitive damage could lead to additional requirements on stable topologies. For example, they could profit from special head topologies or an increased fraction of nodes having nearly the same depth in all stripe trees.

These thoughts and further results in this thesis show that the stability optimization of distribution topologies often leads to conflicts between different notions of stability. Furthermore, in practical applications, additional requirements are posed by technical constraints and efficiency considerations. However, this multiplicity of competing demands does not question the usefulness of our results. In contrast, to be able to find acceptable trade-offs, we must know how the realization of each stability goal is influenced by the properties of a topology. Thus, the study of stable distribution topologies and their characteristics is necessary. Despite possible doubts about their applicability in practical situations, they lead the way towards more resilient and dependable peer-to-peer live streaming systems.

"Only he can make use of favorable winds, who knows where he wants to go to."

— Seneca the Younger

A. Fundamental Inequalities

This appendix lists fundamental inequalities used in the proofs of this thesis.

Lemma A.0.1 ***[BSS09]***
Let $(x_i)_{1\leq i\leq k}$ and $(y_i)_{1\leq i\leq l}$ be two non-increasing sequences of natural numbers, and $x_0, y_0 \in \mathbb{N}$, such that

- $x_0 \geq y_0$,
- $k \geq l$,
- $\sum_{i=0}^{k} x_i \geq \sum_{i=0}^{l} y_i = Y$, and
- $y_i \in \{\lfloor (Y-y_0)/k \rfloor, \lceil (Y-y_0)/k \rceil\}$.

For $0 \leq h \leq l$, it holds that

$$\sum_{i=0}^{k-h} x_i \geq \sum_{i=0}^{l-h} y_i. \tag{A.1}$$

Proof. [BSS09] First, assume that $k = l$. Then, it holds that $0 \leq h \leq k$. For $h = k$, Inequality (A.1) is true since $x_0 \geq y_0$. For $h < k$, we use induction and assume that $\sum_{i=0}^{k-h-1} x_i \geq \sum_{i=0}^{l-h-1} y_i$ holds.

Now, if we had $\sum_{i=0}^{k-h} x_i < \sum_{i=0}^{l-h} y_i$, this would imply $x_{k-h} < y_{l-h}$. By the third condition, we then have $x_{k-h} \leq \lfloor (Y-y_0)/k \rfloor$ and obtain

$$\sum_{i=0}^{k} x_i \leq \sum_{i=0}^{k-h} x_i + h \left\lfloor \frac{Y-y_0}{k} \right\rfloor < \sum_{i=0}^{l-h} y_i + \sum_{i=l-h+1}^{l} y_i = Y. \tag{A.2}$$

However, this contradicts the second condition.

If $k > l$, define $\hat{x}_0 := \sum_{i=0}^{k-l-1} x_i$ and $\hat{x}_i := x_{i+(k-l)}$ for $1 \leq i \leq l$. It holds that $\hat{x}_0 \geq x_0 \geq y_0$. Then, the proposed inequality follows from the case $k = l$. □

Lemma A.0.2 ***Operations on Integer Partitions***
Let $S, k \in \mathbb{N}$ be given and let $t_1, \ldots, t_k \in [0, S]$ be an integer partition of S, i.e. $\sum_{i=1}^{k} t_i = S$. The product $\prod_{i=1}^{k} t_i$ is maximized and the sum $\sum_{i=1}^{k} \binom{t_i}{2}$ is minimized, if it holds that

$$\forall i \in [k] \colon t_i \in \left[\left\lceil \frac{S}{k} \right\rceil, \left\lfloor \frac{S}{k} \right\rfloor \right]. \tag{A.3}$$

Proof. Let $t_1, \ldots, t_k$ be an arbitrary integer partition of S. If there are $a, b \in [k]$ with $t_b - t_a \geq 2$, then set $t'_a := t_a + 1$ and $t'_b := t_b - 1$. Substituting t_a, t_b by t'_a, t'_b will increase the product value:

$$\prod_{i\in[k]} t_i < \prod_{i\in[k]} t_i + \left(\prod_{i\in[k]\setminus\{a,b\}} t_i\right)(t_b - t_a - 1) \tag{A.4}$$

$$= \left(\prod_{i\in[k]\setminus\{a,b\}} t_i\right) \cdot (t_a t_b + t_b - t_a - 1) \tag{A.5}$$

$$= \left(\prod_{i\in[k]\setminus\{a,b\}} t_i\right) \cdot (t'_a)(t'_b) \tag{A.6}$$

Furthermore, it holds that

$$\sum_{i\in[k]} \binom{t_i}{2} = \frac{t_a(t_a-1)}{2} + \frac{t_b(t_b-1)}{2} + \sum_{i\in[k]\setminus\{a,b\}} \binom{t_i}{2} \tag{A.7}$$

$$> \frac{t_a(t_a-1) + t_b(t_b-1) - 2(t_b - t_a - 1)}{2} + \sum_{i\in[k]\setminus\{a,b\}} \binom{t_i}{2} \tag{A.8}$$

$$= \frac{(t_a+1)(t_a)}{2} + \frac{(t_b-1)(t_b-2)}{2} + \sum_{i\in[k]\setminus\{a,b\}} \binom{t_i}{2} \tag{A.9}$$

$$= \binom{t'_a}{2} + \binom{t'_b}{2} + \sum_{i\in[k]\setminus\{a,b\}} \binom{t_i}{2} \tag{A.10}$$

The transformation assures that $t_a < t'_a \leq t'_b < t_b$. In particular, it decreases $\max(|t_a - S/k|, |t_b - S/k|)$ with each step. Iterating this process constantly increases and decreases the value of the considered product and sum, respectively. The process stops as soon as Property (A.3) is reached. □

Lemma A.0.3 ***Sums over Ordered Partitions***
Let $k \in \mathbb{N}$ and $t_1, \ldots, t_k, S \in \mathbb{R}^+$ with $t_1 \leq t_2 \leq \ldots \leq t_k$ and $\sum_{i\in[k]} t_i = S$ be given. It holds that

$$\forall z \in [k]\colon \sum_{i=1}^{z} t_i \leq \frac{z}{k} S.$$

Proof. Fix an arbitrary $z \in [k]$. If $t_z \leq \frac{S}{k}$, we clearly have $\sum_{i=1}^{z} t_i \leq z\frac{S}{k}$.

Otherwise, it holds that $t_{z+1}, \ldots, t_k \geq \frac{S}{k}$. This leads to

$$\sum_{i=1}^{z} t_i = S - \sum_{i=z+1}^{k} t_i \leq S - (k-z)\frac{S}{k} = \frac{z}{k} S. \tag{A.11}$$

□

B. The Distance Distribution of MDS Codes

The available literature on MDS codes (e.g., [MS93, Bie05, Rot06, BBF+06, PWB11]) has a heavy focus on *linear* MDS codes. Due to this reason, all sources known to the author only cover the distance distribution for this subclass. For the use in Section 5.3.3, in this appendix we show that the distance distribution for linear MDS codes actually holds for *all* MDS codes.

The following is a well-known result, e.g., it can be obtained from [MS93, Theorem 11.6] and Corollary 5.3.30. Let M be a linear, C-ary MDS code of length k and dimension t. Such a code has minimum distance $d = k - t + 1$. For every $\mathbf{v} \in M$, the distance distribution $\mathbf{d}_{C,k,t}(\mathbf{v})$ of $\mathbf{v}$ in M is given by

$$\mathbf{d}_{C,k,t}(\mathbf{v})_i = \begin{cases} 1 & \text{, if } i = 0 \\ 0 & \text{, if } 0 < i < k - t + 1 \\ \binom{k}{i} \sum_{r=0}^{i-d} (-1)^r \binom{i}{r} (C^{i-d+1-r} - 1) & \text{, if } k - t + 1 \leq i \leq k. \end{cases} \tag{B.1}$$

We show that this distance distribution holds for non-linear MDS codes as well.

Lemma B.0.4
Every C-ary MDS code M of C^t codewords and length k is *distance-invariant* and the distance distribution for every $\mathbf{v} \in M$ is $\mathbf{d}_{C,k,t}(\mathbf{v})$.

Proof. Due to the Lemmata 5.3.31 and 5.3.35, M has dual distance $d^\perp = t + 1$. Since M has minimum distance $d = k - t + 1$, for every $\mathbf{v} \in M$, the distance distribution $\mathbf{d}(\mathbf{v})$ in M has non-zero entries for at most t distances greater 0. Since this number is smaller than $d^\perp$, according to [MS93, Theorem 6.6], M is distance-invariant.

Now, we show the following claim.

Claim B.0.5
Let M be given from Lemma B.0.4. For any $\mathbf{v} \in M$, the maximum possible number of codewords of M that are in distance i from $\mathbf{v}$ is $\mathbf{d}_{C,k,t}(\mathbf{v})_i$.

Proof. The proof is inspired by the proofs on the weight distribution of linear MDS codes in [PWB11, Chapter 4.4].

Since M is distance-invariant, fix an arbitrary $\mathbf{v} \in M$. For $I \subseteq [k]$ with $i = |I|$ and $i \in [k]$, the maximum number of codewords in $M \setminus \{\mathbf{v}\}$ that *coincide* with $\mathbf{v}$ in all positions I is $B_I := \lceil C^{t-i} \rceil - 1$:

- Codewords coinciding in at least t positions can have at most distance $k - t < d$. However, M has minimum distance $d = k - t + 1$. We obtain $B_I = 0$ for $i \geq t$.
- Since M has dual distance $d^\perp = t + 1$, due to Lemma 5.3.31, its matrix representation is an $\mathrm{OA}(C^t, k, C, t)$ with index 1. Therefore, for every $i < t$, it is an $\mathrm{OA}(C^t, k, C, i)$ of index C^{t-i} and there are exactly C^{t-i} rows of M with the same combination of letters in the columns I. Subtracting 1 for $\mathbf{v}$ itself gives B_I.

We sum up B_I for all $I \subseteq [k]$ with $i = |I|$ as $B_i := \binom{k}{i}(\lceil C^{t-i} \rceil - 1)$. It holds that

$$B_i = \sum_{j=k-t+1}^{k-i} \binom{k-j}{i} A_j, \tag{B.2}$$

where A_j is the maximum possible number of codewords of M in distance j from $\mathbf{v}$:

- Equation (B.2) is true for $i \geq t$, since in this case the sum will be empty.
- For $i < t$, every existing $\mathbf{w} \in M$ with a distance $j \leq k - i$ from $\mathbf{v}$ has to coincide on $k - j \geq i$ positions with $\mathbf{v}$. Thus, it is counted $\binom{k-j}{i}$ times (once for every possible choice of I as a subset of their coinciding positions) for B_i. The maximality of the A_j follows from the maximality of B_i.

The A_j will be zero for $j < d = k - t + 1$ and the remaining A_j can be computed using Equation (B.2). They exactly resemble the corresponding entries in the distance- (and weight-) distribution of a linear MDS code with length k and dimension t (cmp. Equation B.1):

$$A_j = \begin{cases} \binom{k}{j} \sum_{r=0}^{j-d} (-1)^r \binom{j}{r} (C^{j-d+1-r} - 1) & \text{, if } d \leq j \\ 0 & \text{, if } 1 \leq j < d \end{cases} \tag{B.3}$$

□

The entries of $\mathbf{d}_{C,k,t}(\cdot)$ sum up to C^t. Hence, due to Claim B.0.5, each $\mathbf{v} \in M$ must have distance distribution $\mathbf{d}_{C,k,t}(\mathbf{v})$. Otherwise, M had less than C^t codewords. □

C. Index of Notation

Numbers, Sets & Concepts

$[a, b]$	integer interval $\{a, a+1, \ldots, b\}$	22
$[a]$	integer interval $[1, a]$	22
$H(n)$	n-th harmonic number	43
$\mathbf{x}, \mathbf{v}$	vectors	111
$\mathcal{P}(X)$	power set of X	37
$R_O(x, y)$	approximation ratio	38

(Multi-)Graphs

u, v, w	nodes	23
X, Y, Z	sets of nodes (*also:* attacks)	23
$m_G(X, Y)$	multiplicity	23
$m_G(v)$	multiplicity of node v	23
$G[X]$	submultigraph induced by X	23
$e_G(X)$	number of edges in $G[X]$	23

Distribution Topologies

$\mathcal{T}, \mathcal{C}, \mathcal{D}$	distribution topologies	23
$\mathbb{T}(n, \mathrm{c}, k)$	class of bandwidth-restricted distribution topologies	27
$\mathbb{T}(n, C, k)$	class of source-bandwidth-restricted distribution topologies	27
$d_T(v)$	depth of node v in tree T	24
$d(T), d(\mathcal{T})$	depth of a tree or topology	24
$L_i(T), L_i(\mathcal{T})$	depth level i of a tree or topology	25
$H_i^{\mathcal{T}}$	set of heads of stripe i in topology $\mathcal{T}$	25
$H^{\mathcal{T}}$	set of all heads of topology $\mathcal{T}$	25
$\mathrm{pred}_i^{\mathcal{T}}(v)$	predecessor set of v in stripe i of $\mathcal{T}$	25
$\mathrm{succ}_i^{\mathcal{T}}(X)$	successor set of node set X in stripe i of $\mathcal{T}$	26
$\mathrm{parent}_i^{\mathcal{T}}(v)$	parent of node v in stripe i of $\mathcal{T}$	26
$\mathrm{child}_i^{\mathcal{T}}(v)$	children of v in stripe i of $\mathcal{T}$	26
$\mathrm{succ}_i^{\mathcal{T}\rightarrow}(X)$	forward successor set of node set X in stripe i of $\mathcal{T}$	26

Attacks & Damage

$\mathrm{inc}_X^{\mathcal{T}}(v)$	number of intact $s \to v$ paths in $\mathcal{T}$ despite removal of X	33
$L_{X,z}^{\mathrm{MDC}}$	Lost Service Set under Multiple Description Coding	33
$L_{X,z}^{\mathrm{FEC}}$	Lost Service Set under Forward Error Correction	34
$a^{\mathcal{T}}(X)$	LiSS-damage / packet loss	32
$\mathrm{b}^{\mathcal{T}}(X,z)$	LoSS-damage	33
$\mathrm{bec}^{\mathcal{T}}(X,z)$	FEC-LoSS-damage	34
$\mathrm{bf}^{\mathcal{T}}(X,z)$	forward-damage	102
$\delta_i^{C,k}$	damage sequence	56
σ_i^C	edge sequence	72
$D(\mathcal{H})$	dependency graph of head topology $\mathcal{H}$	69
$\mathcal{H}_i$	subtopology of head topology $\mathcal{H}$	73

Analysis of Forward-Stable Topologies

$\chi(\mathcal{T},t)$	set of t-restricted attacks on $\mathcal{T}$	104
$\mathrm{OA}(n,k,C,t)$	Orthogonal Array	111
$\mathrm{PA}(n,k,C,t)$	Packing Array	124
M	matrix or code	127
m_{vi}	entry in row v and column i of a matrix M	109
$M^{\mathcal{T}}$	matrix of forward successor sets of $H^{\mathcal{T}}$	109
$\sigma_i(H_i^{\mathcal{T}})$	bijection between $H_i^{\mathcal{T}}$ and $\lvert H_i^{\mathcal{T}}\rvert$ used to define $M^{\mathcal{T}}$	109
$\sigma(X)$	vector attack for attack X	109
$\mathfrak{X},\mathfrak{Y},\mathfrak{Z}$	vector attacks	114
$\mathrm{d}(\mathbf{v},\mathbf{x})$	Hamming distance between $\mathbf{v}$ and $\mathbf{x}$	110
$\mathbf{s}$	head distribution	113
$\mathbf{a}$	attack distribution	113
$\mathbb{V}(\mathbf{s})$	head vector space for head distribution $\mathbf{s}$	113
$\mathbb{X}(\mathbf{a},\mathbf{s})$	vector attacks with distribution $\mathbf{a}$ for head distribution $\mathbf{s}$	113
$N_d(\mathfrak{X},\mathbf{s})$	vectors from $\mathbb{V}(\mathbf{s})$ in Hamming Distance d from $\mathfrak{X}$	114

Analysis of Failure-Stable Topologies

$A_{x,v}^{\mathcal{T}}$	random variable for $k - \mathrm{inc}_X(v)$ in case of x failing nodes	142
$A_x^{\mathcal{T}}$	random variable for $a^{\mathcal{T}}(X)$ in case of x failing nodes	142

D. List of Figures

E. Bibliography

[ACK+00] G. Ausiello, P. Crescenzi, V. Kann, Marchetti-Sp, Giorgio Gambosi, and Alberto M. Spaccamela. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 2000.

[Aka] The Telegraph: Royal wedding tops iTunes TV chart. http://www.telegraph.co.uk/news/uknews/royal-wedding/8490562/Royal-wedding-tops-iTunes-TV-chart.html. retrieved 01-18-2012.

[AMS06] Noga Alon, Dana Moshkovitz, and Shmuel Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2:153–177, 2006.

[ASSC02] Ian F. Akyildiz, W. Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[BB12] Robert F. Bailey and Andrea C. Burgess. Generalized packing designs (Preprint). *Discrete Mathematics*, 2012.

[BBF+06] Anton Betten, Michael Braun, Harald Fripertinger, Adalbert Kerber, Axel Kohnert, and Alfred Wassermann. *Error-Correcting Linear Codes - Classification by Isometry and Applications*, volume 18 of *Algorithms and Computation in Mathematics*. Springer-Verlag Berlin Heidelberg, 2006.

[BBG+09] Andreas Brieg, Michael Brinkmeier, Sascha Grau, Mathias Fischer, and Guenter Schaefer. Attacker Independent Stability Guarantees for Peer-2-Peer-Live-Streaming Topologies. In *Second International Conference on Communication Theory, Reliability, and Quality of Service*, pages 20–25. IEEE, 2009.

[BBK02] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. Scalable application layer multicast. *SIGCOMM Comput. Commun. Rev.*, 32:205–217, 2002.

[BFGS09a] Michael Brinkmeier, Mathias Fischer, Sascha Grau, and Guenter Schaefer. Towards the design of unexploitable construction mechanisms for multiple-tree based P2P streaming systems. In *Kommunikation in Verteilten Systemen (KiVS)*, pages 193–204. Springer, 2009.

[BFGS09b] Michael Brinkmeier, Mathias Fischer, Sascha Grau, and Thorsten Strufe. Methods for Improving Resilience in Communication Networks and P2P Overlays. In Otto Spaniol, editor, *Praxis der Informationsverarbeitung und Kommunikation, PIK*, volume 32, pages 64–78. K. G. Saur Publishing, 2009.

[Bie96] Jürgen Bierbrauer. Construction of orthogonal arrays. *Journal of Statistical Planning and Inference*, 56(1):39 – 47, 1996.

[Bie05] Jürgen Bierbrauer. *Introduction to Coding Theory.* Chapman & Hall / CRC, 2005.

[BLB+04] Stefan Birrer, Dong Lu, Fabián E. Bustamante, Yi Qiao, and Peter A. Dinda. FatNemo: Building a Resilient Multi-source Multicast Fat-Tree. In Chi-Hung Chi, Maarten van Steen, and Craig E. Wills, editors, *Web Content Caching and Distribution: 9th International Workshop, WCW 2004, Beijing, China, October 18-20, 2004. Proceedings*, volume 3293 of *Lecture Notes in Computer Science*, pages 182–196. Springer, 2004.

[BLS99] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey.* SIAM, Philadelphia, PA, USA, 1999.

[Bri08] Andreas Brieg. Klassifikation optimal stabiler Live-Streaming Topologien (Classification of Optimally Stable Live-Streaming-Topologies). Diploma Thesis, Ilmenau University of Technology, 2008.

[BSS09] Michael Brinkmeier, Guenter Schaefer, and Thorsten Strufe. Optimally DoS Resistant P2P Topologies for Live Multimedia Streaming. *IEEE Transactions on Parallel and Distributed Systems*, 20(6):831–844, 2009.

[Bus52] K.A. Bush. Orthogonal Arrays of Index Unity. *Ann. Math. Statist.*, 23:426–434, 1952.

[CD06] Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications).* Chapman & Hall/CRC, 2006.

[CDK+03] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, Animesh Nandi, Antony Rowstron, and Atul Singh. Splitstream: high-bandwidth multicast in cooperative environments. *SIGOPS Oper. Syst. Rev.*, 37:298–313, 2003.

[CDKM00] Robert D. Carr, Srinivas Doddi, Goran Konjevod, and Madhav Marathe. On the red-blue set cover problem. In *Proceedings of the eleventh annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 345–353, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[CDKR02] M. Castro, P. Druschel, A.-M. Kermarrec, and A.I.T. Rowstron. Scribe: a large-scale and decentralized application-level multicast infrastructure. *Selected Areas in Communications, IEEE Journal on*, 20(8):1489 – 1499, 2002.

[CGM83] P.M. Camerini, G. Galbiati, and F. Maffioli. On the complexity of finding multi-constrained spanning trees. *Discrete Applied Mathematics*, 5(1):39–50, 1983.

[CH96] Ioana Constantinescu and Werner Heise. On the Concept of Code-Isomorphy. *Journal of Geometry*, 57(1-2):63–69, 1996.

[Cha03] Yatin Chawathe. Scattercast: an adaptable broadcast distribution framework. *Multimedia Syst.*, 9:104–118, 2003.

[Cre97] P. Crescenzi. A short guide to approximation preserving reductions. In *Computational Complexity, 1997. Proceedings., Twelfth Annual IEEE Conference on (Formerly: Structure in Complexity Theory Conference)*, pages 262–273, 1997.

[CRSZ02] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan, and Hui Zhang. A Case for End System Multicast. *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast*, 20(8), 2002.

[DB02] Wiebke S. Diestelkamp and Jay H. Beder. On the decomposistion of orthogonal arrays. *Utilitas Mathematica*, (61):65–86, 2002.

[Dee92] Stephen Edward Deering. *Multicast routing in a datagram internetwork*. PhD thesis, Stanford University, Stanford, CA, USA, 1992.

[DF10] György Dán and Viktória Fodor. Stability and performance of overlay multicast systems employing forward error correction. *Perform. Eval.*, 67:80–101, 2010.

[DFK06] György Dán, Viktória Fodor, and Gunnar Karlsson. On the stability of end-point-based multimedia streaming. In *NETWORKING'06*, Lecture Notes in Computer Science, pages 678–690, Berlin, Heidelberg, 2006. Springer-Verlag.

[DHRS07] Prithula Dhungel, Xiaojun Hei, Keith W. Ross, and Nitesh Saxena. The pollution attack in p2p live video streaming: measurement results and defenses. In *Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*, P2P-TV '07, pages 323–328, New York, NY, USA, 2007. ACM.

[Die05] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.

[DS04] Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5):247–254, 2004.

[FDGS11] Mathias Fischer, Sebastian Delling, Sascha Grau, and Guenter Schaefer. Underlay-Robust Application Layer Multicast (Extended Abstract). In *International Performance Computing and Communications Conference*, IPCCC, pages 1–2, Orlando, Florida, 2011. IEEE.

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[FGKS11] Mathias Fischer, Sascha Grau, Sebastian Kehr, and Guenter Schaefer. Attack-Resilient and Multiple-Tree-based P2P-IPTV Distribution. In *International Congress of Ultra Modern Telecommunications and Control Systems*, ICUMT, pages 1–8, Budapest, Hungary, 2011. IEEE.

[Fis12] Mathias Fischer. *Construction of Attack-Resilient and Efficient Overlay-Topologies for Large-Scale P2P-based IPTV Infrastructures*. PhD thesis, Ilmenau University of Technology, 2012.

[FKGS11] Mathias Fischer, Michael Kissmann, Sascha Grau, and Guenter Schaefer. On Virtualization-based Network Support for Peer-assisted Live-Streaming Applications. In *Network of the Future*, NoF, pages 25–30, Paris, France, 2011. IEEE.

[FSK05] Bryan Ford, Pyda Srisuresh, and Dan Kegel. Peer-to-peer communication across network address translators. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC '05, pages 13–13, Berkeley, CA, USA, 2005. USENIX Association.

[FY07] Zongming Fei and Mengkun Yang. A proactive tree recovery mechanism for resilient overlay multicast. *Networking, IEEE/ACM Transactions on*, 15(1):173 –186, 2007.

[GA04] Meng Guo and Mostafa H. Ammar. Scalable Live Video Streaming to Cooperative Clients Using Time Shifting and Video Patching. In *Proc. IEEE INFOCOM*, 2004.

[GCM11] Gabriela Gheorghe, Renato Lo Cigno, and Alberto Montresor. Security and privacy issues in p2p streaming systems: A survey. *Peer-to-Peer Networking and Applications*, 4(2):75–91, 2011.

[GFBS11] Sascha Grau, Mathias Fischer, Michael Brinkmeier, and Günter Schäfer. On Complexity and Approximability of Optimal DoS Attacks on Multiple-Tree P2P Streaming Topologies. *IEEE Transactions on Dependable and Secure Computing*, 8(2):270–281, 2011.

[GFS11] Sascha Grau, Mathias Fischer, and Günter Schäfer. On the Dependencies between Source Neighbors in Optimally DoS-stable P2P Streaming Topologies. In *IEEE International Conference on Distributed Computing Systems 2011*, ICDCS, pages 121–130, Minneapolis, MN, 2011. IEEE Computer Society.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, 1979.

[GJKJ00] David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole Jr. Overcast: Reliable multicasting with an overlay network. In *Proc. Usenix Fourth Symposium on Operating System Design and Implementation*, 2000.

[Gon07] Teofilo F. Gonzalez. *Handbook of Approximation Algorithms and Metaheuristics (Chapman & Hall/Crc Computer & Information Science Series)*. Chapman & Hall/CRC, 2007.

[Goy01] V.K. Goyal. Multiple description coding: compression meets the network. *Signal Processing Magazine, IEEE*, 18(5):74 –93, 2001.

[Gra12] Sascha Grau. Attack-Resilient Multitree Data Distribution Topologies. In *16th International Conference On Principles Of DIstributed Systems*, OPODIS 2012, LNCS 7702, pages 196–208, Rome, Italy, 2012. Springer Berlin Heidelberg.

[Gum11] Wolfgang Gummlich. Experimenteller Vergleich exakter Algorithmen für ein Angreiferproblem (Experimental Evaluation of Exact Algorithms for an Attacker Problem), 2011. Student Research Project, Ilmenau University of Technology.

[GZL03] Jiang Guo, Ying Zhu, and Baochun Li. Codedstream: Live media streaming with overlay coded multicast. In *In Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking*, MMCN, pages 28–39, 2003.

[HA07] M.T. Helmick and F.S. Annexstein. Depth-Latency Tradeoffs in Multicast Tree Algorithms. In *Advanced Information Networking and Applications, 2007. AINA '07. 21st International Conference on*, pages 555 –564, 2007.

[Hoc97] Dorit S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston, MA, USA, 1997.

[Hol10] Vincent Holluba. Untersuchung eines Bandbreitenbasierten Angriffsmodells für Peer to Peer-Streaming-Systeme (Evaluation of a Bandwidth-Based Attack Model for Peer-to-Peer Streaming Systems). Bachelor Thesis, Ilmenau University of Technology, 2010.

[HSS99] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal Arrays: Theory and Applications*. Springer-Verlag, New York, 1999.

[HvR08] Maya Haridasan and Robbert van Renesse. SecureStream: An intrusion-tolerant protocol for live-streaming dissemination. *Computer Communications*, 31(3):563–575, 2008.

[Kan92] V. Kann. *On the Approximability of NP-complete Optimization Problems*. PhD thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.

[KAS10] Lachezar Krumov, Adriana Andreeva, and Thorsten Strufe. Resilient peer-to-peer live-streaming using motifs. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pages 1–8, 2010.

[KLS03] J. Könemann, A. Levin, and A. Sinha. Approximating the degree-bounded minimum diameter spanning tree problem. In *Algorithmica*, pages 109–121. Springer, 2003.

[KMSV99] Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh Vazirani. On Syntactic versus Computational Views of Approximability. *SIAM J. Comput.*, 28:164–191, 1999.

[Kol00] Stavros Kolliopoulos. Approximating Covering Integer Programs with Multiplicity Constraints. *Discrete Appl. Math*, 129:461–473, 2000.

[KR03] Daniel Kobler and Udi Rotics. Finding maximum induced matchings in subclasses of claw-free and P5-free graphs, and in graphs with matching and induced matching of equal maximum size. *Algorithmica*, 37(4):327–346, 2003.

[KSU11] Oh Chan Kwon, Hwangjun Song, and Tai-Won Um. Overlay multicast tree construction algorithm for stable multimedia service. In *Consumer Communications and Networking Conference (CCNC)*, pages 342–346. IEEE, 2011.

[LCC+11] Bo Liu, Yanchuan Cao, Yi Cui, Yuan Xue, Fan Qiu, and Yansheng Lu. Minimizing service disruption in peer-to-peer streaming. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 1066 –1071, 2011.

[LMS+97] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. Practical loss-resilient codes. In *Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing*, STOC '97, pages 150–159, New York, NY, USA, 1997. ACM.

[LMSW07] Thomas Locher, Remo Meier, Stefan Schmid, and Roger Wattenhofer. Push-to-pull peer-to-peer live streaming. In *Distributed Computing (DISC)*, volume 4731 of *Lecture Notes in Computer Science*, pages 388–402. Springer, 2007.

[MFL00] Chang-Xing Ma, Kai-Tai Fang, and Erkki Liski. A new approach in constructing orthogonal and nearly orthogonal arrays. *Metrika*, 50:255–268, 2000.

[MGM06] Sergio Marti and Hector Garcia-Molina. Taxonomy of Trust: Categorizing P2P Reputation Systems. *Computer Networks*, 50(4):472 – 484, 2006. Management in Peer-to-Peer Systems.

[MM99] Terry A. McKee and F. R. McMorris. *Topics in Intersection Graph Theory*. SIAM, Philadelphia, PA, USA, 1999.

[MR07] N. Magharei and R. Rejaie. PRIME: Peer-to-Peer Receiver-drIven MEsh-Based Streaming. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, pages 1415 –1423. IEEE, 2007.

[MS93] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Mathematical Library, 1993.

[NL08] Nam-Ky Nguyen and Min-Quan Liu. An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays. *Computational Statistics and Data Analysis*, 52:5269–5276, 2008.

[OD85] Ulrich Oberst and Arne Dür. A constructive characterization of all optimal linear codes. In *séminaire d'algèbre Paul Dubreil et Marie-Paule Malliavin*, volume 1146/1985 of *Lecture Notes in Mathematics*, pages 176–213, 1985.

[OSV09] Christian Ortolf, Christian Schindelhauer, and Arne Vater. Classifying peer-to-peer network coding schemes. In *Proceedings of the twenty-first annual Symposium on Parallelism in Algorithms and Architectures*, SPAA '09, pages 310–318, New York, NY, USA, 2009. ACM.

[PPK10] Kunwoo Park, Sangheon Pack, and Ted Kwon. An adaptive peer-to-peer live streaming system with incentives for resilience. *Computer Networks*, 54(8):1316 – 1327, 2010.

[PPt] PPLive. http://www.pptv.com.

[PWB11] Ruud Pellikaan, Xin-Wen Wu, and Stanislav Bulygin. *Error-Correcting Codes and Cryptology (Manuscript)*. Cambridge University Press, 2011.

[PWC03] Venkata N. Padmanabhan, Helen J. Wang, and Philip A. Chou. Resilient Peer-to-Peer Streaming. In *Network Protocols, IEEE International Conference on*, page 16, Atlanta, Georgia, USA, 2003. IEEE Computer Society.

[PWCS02] Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou, and Kunwadee Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proceedings of the 12th international workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '02, pages 177–186, New York, NY, USA, 2002. ACM.

[Rot06] Ron M. Roth. *Introduction to Coding Theory*. Cambridge University Press, 2006.

[Rou73] Nicholas D. Roussopoulos. A max {m,n} algorithm for determining the graph H from its line graph G. *Inf. Process. Lett.*, 2(4):108–112, 1973.

[RS60] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. SIAM*, 8:300–304, 1960.

[RS11] Dror Rawitz and Shimon (Moni) Shahar. Partial multicovering and the d-consecutive ones property. *Discrete Optimization*, 8(4):555–567, 2011.

[RSS07] Michael Rossberg, Guenter Schaefer, and Thorsten Strufe. Using recurring costs for reputation management in peer-to-peer streaming systems. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 283–292, Nice, France, 2007. IEEE.

[Rö10] Johannes Röckert. Optimierung von Peer-To-Peer Streamingtopologien bezüglich ihrer Robustheit bei Ausfall randomisiert gewählter Knotenmengen (Optimization of Peer-To-Peer Streaming Topologies with Respect to Robustness Against the Failure of Randomly Chosen Node Sets), 2010. Diploma Thesis, Ilmenau University of Technology.

[Seg55] B. Segre. Curve razionali normali e k-archi negli spazi finiti. *Ann. Math. Pura Appl.*, (39):357–359, 1955.

[SFC08] Thomas Silverston, Olivier Fourmaux, and Jon Crowcroft. Towards an incentive mechanism for peer-to-peer multimedia live streaming systems. In *Proceedings of the 2008 Eighth International Conference on Peer-to-Peer Computing*, P2P '08, pages 125–128, Washington, DC, USA, 2008. IEEE Computer Society.

[SIB12] Raymond Sweha, Vatche Ishakian, and Azer Bestavros. Angelcast: cloud-based peer-assisted live streaming using optimized multi-tree construction. In *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, pages 191–202, New York, NY, USA, 2012. ACM.

[Sil60] Robert Silverman. A metrization for power-sets with applications to combinatorial analysis. *Canad. J. Math.*, 12:158–176, 1960.

[SL07] Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of Computing*, STOC '07, pages 661–670, New York, NY, USA, 2007. ACM.

[Sla97] Petr Slavík. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64(5):251–254, 1997.

[SM02] Brett Stevens and Eric Mendelsohn. Packing arrays and packing designs. *Designs, Codes and Cryptography*, 27:165–176, 2002.

[SMZ04] Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang. An analysis of live streaming workloads on the internet. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 41–54, New York, NY, USA, 2004. ACM.

[SSNRR10] Jeff Seibert, Xin Sun, Cristina Nita-Rotaru, and Sanjay Rao. Towards securing data delivery in peer-to-peer streaming. In *Proceedings of the 2nd international conference on COMmunication systems and NETworks*, COMSNETS'10, pages 327–336, Piscataway, NJ, USA, 2010. IEEE Press.

[Str07] Thorsten Strufe. *Ein Peer-To-Peer basierter Ansatz für die Live-Übertragung multimedialer Datenströme (A Peer-to-Peer Based Approach for the Live Distribution of Multimedia Data Streams)*. PhD thesis, Ilmenau University of Technology, 2007.

[THD03] D. A. Tran, K. A. Hua, and T. Do. ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1283–1292 vol.2. IEEE, 2003.

[TJ07] G. Tan and S.A. Jarvis. Improving the Fault Resilience of Overlay Multicast for Media Streaming. *Parallel and Distributed Systems, IEEE Transactions on*, 18(6):721 –734, 2007.

[TJD+05] Xuping Tu, Hai Jin, Dafu Deng, Chao Zhang, and Quan Yuan. Design and Deployment of Locality-Aware Overlay Multicast Protocol for Live Streaming Services. In Hai Jin, Daniel Reed, and Wenbin Jiang, editors, *Network and Parallel Computing*, volume 3779 of *Lecture Notes in Computer Science*, pages 105–112. Springer Berlin / Heidelberg, 2005.

[TWSN08] Ye Tian, Di Wu, Guangzhong Sun, and Kam-Wing Ng. Improving stability for peer-to-peer multicast overlays by active measurements. *Journal of Systems Architecture*, 54(1–2):305 – 323, 2008.

[Vaz04] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2004.

[VS10] Constantinos Vassilakis and Ioannis Stavrakakis. Minimizing node churn in peer-to-peer streaming. *Comput. Commun.*, 33(14):1598–1614, 2010.

[VYF06] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous Unstructured Tree-Based Peer-to-Peer Multicast. In *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, pages 2 –11, 2006.

[Weg05] Ingo Wegener. *Complexity theory - exploring the limits of efficient algorithms*. Springer, 2005.

[WL07] Mea Wang and Baochun Li. Lava: A reality check of network coding in peer-to-peer live streaming. In *INFOCOM*, pages 1082–1090. IEEE, 2007.

[WLX08] Feng Wang, Jiangchuan Liu, and Yongqiang Xiong. Stable Peers: Existence, Importance, and Application in Peer-to-Peer Live Video Streaming. In *INFOCOM 2008. The 27th Conference on Computer Communications*, pages 1364 –1372, 2008.

[WLX11] Feng Wang, Jiangchuan Liu, and Yongqiang Xiong. On Node Stability and Organization in Peer-to-Peer Video Streaming Systems. *IEEE Systems Journal*, 5(4):440–450, 2011.

[WLZ08] Chuan Wu, Baochun Li, and Shuqiao Zhao. Exploring large-scale peer-to-peer live streaming topologies. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4:19:1–19:23, 2008.

[Wol82] L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.

[WSHW08] Arno Wacker, Gregor Schiele, Sebastian Holzapfel, and Torben Weis. A NAT Traversal Mechanism for Peer-To-Peer Networks. In *Proceedings of the 8th International Conference on Peer-to-Peer Computing, IEEE P2P'08*, Aachen, Germany, 2008.

[WXL10] Feng Wang, Yongqiang Xiong, and Jiangchuan Liu. mtreebone: A collaborative tree-mesh overlay network for multicast video streaming. *IEEE Transactions on Parallel and Distributed Systems*, 21:379–392, 2010.

[WXZJ06] Wenjie Wang, Yongqiang Xiong, Qian Zhang, and Sugih Jamin. Ripplestream: Safeguarding p2p streaming against dos attacks. *Multimedia and Expo, IEEE International Conference on*, 0:1417–1420, 2006.

[Xu02] Hongquan Xu. An Algorithm for Constructing Orthogonal and Nearly Orthogonal Arrays with Mixed Levels and Small Runs. *Technometrics*, 44:356–368, 2002.

[XZ06] Liang Xie and Sencun Zhu. Message Dropping Attacks in Overlay Networks: Attack Detection and Attacker Identification. In *Securecomm and Workshops, 2006*, pages 1 –10, 2006.

[Yan78] Mihalis Yannakakis. Node-and edge-deletion NP-complete problems. In *Proceedings of the tenth annual ACM symposium on Theory of Computing*, STOC '78, pages 253–264, New York, NY, USA, 1978. ACM.

[YLY+04] Hao Yang, Haiyun Luo, Yang Yi, Songwu Lu, and Lixia Zhang. HOURS: Achieving DoS Resilience in an Open Service Hierarchy. In *The International Conference on Dependable Systems and Networks (DSN)*, Florence, Italy, 2004.

[Zat] Zattoo. http://www.zattoo.com.

[ZLLY05] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Y.-S.P. Yum. CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2102 – 2111 vol. 3, 2005.

[ZYL+07] Sencun Zhu, Chao Yao, Donggang Liu, Sanjeev Setia, and Sushil Jajodia. Efficient security mechanisms for overlay multicast based content delivery. *Comput. Commun.*, 30(4):793–806, 2007.

[ZZSY07] Meng Zhang, Qian Zhang, Lifeng Sun, and Shiqiang Yang. Understanding the Power of Pull-Based Streaming Protocol: Can We Do Better? *IEEE Journal on Selected Areas in Communications*, 25(9):1678–1694, 2007.